COPPEAD/UFRJ

RELATÓRIO COPPEAD Nº 61

A FORMALIZAÇÃO DO PROBLEMA DE NIVELAMENTO DE RECURSOS EM REDES DE PLANEJAMENTO DE GRANDE PORTE E SUA SOLUÇÃO ÓTIMA - EFICIENTE

Jorge Alberto García Gómez*

Junho 1981

* Doutor em Ciências pela COPPE/UFRJ e professor adjunto do programa de Mestrado em Administração de Empresas da COPPE/UFRJ.
O autor agradece o suporte financeiro dado pelo CNPq e pela IBM do Brasil, sem o qual esta pesquisa não poderia ter sido realizada.

I. INTRODUÇÃO

Neste trabalho, procura-se desenvolver uma técnica para a solução do problema de nivelamento de recursos em uma rede de projeto. A intenção é apenas de nos concentrarmos no problema de nivelamento de recursos, buscando desenvolver um método de planejamento satisfatório, viável em termos computacionais e de fácil aplicação na vida real.

Mais especificamente, nesta pesquisa, objetivamos apresentar, basicamente, duas contribuições à teoria do conhecimento científico correlato ao planejamento em redes. A primeira delas é a formalização das premissas básicas que definem o problema de nivelamento de recursos em redes de planejamento que, até a data presente, não foi formalizado pelos estudiosos do assunto. A segunda contribuição diz respeito à apresentação, neste mesmo trabalho, de um modelo matemático cujos esforços convirjam para o problema de nivelamento de recursos, como definido formalmente. Este modelo matemático, oriundo da programação heurística, preenche os quesitos de admissibilidade e consistência, garantindo assim a otimilidade das soluções fornecidas pelo mesmo, no sentido de Hart et alii [5].

No algorítmo proposto procura-se, através de uma busca sistemática, encontrar um perfil de programação das atividades de um projeto, visando a melhor distribuir os recursos financeiros (fluxo de caixa) ao longo do tempo.

II. ANTECEDENTES DO PROBLEMA

Uma das maiores vantagens da representação de um projeto através de redes é a possibilidade de serem facilmente geradas as informações das necessidades de recursos no decorrer do tempo. Estas informações são obtidas como um subproduto do método tradicional de computação do caminho crítico. O único requisito desta representação é as demandas de recursos associadas a cada atividade serem apresentadas separadamente.

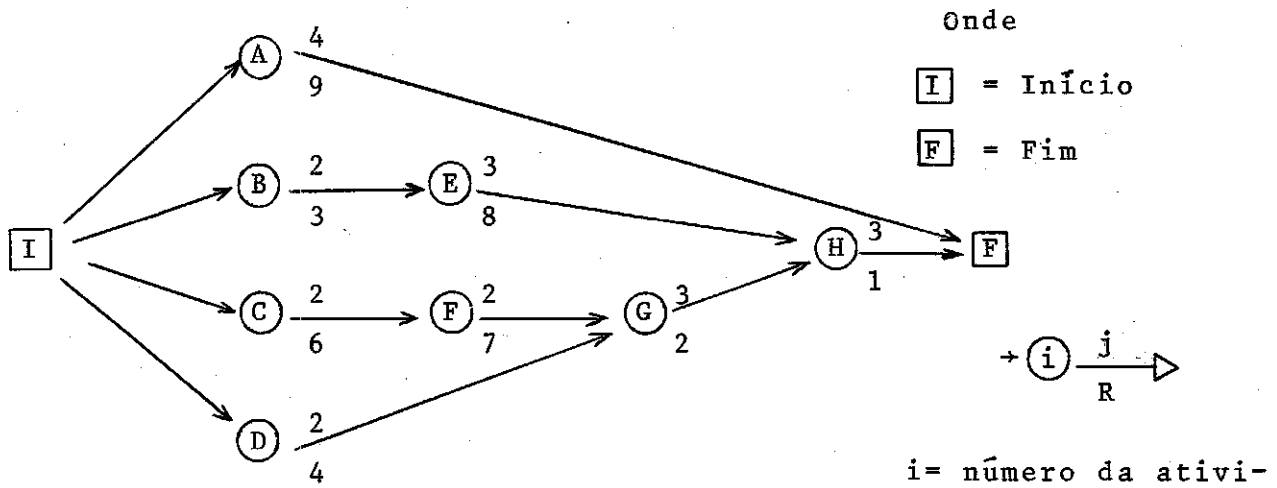


Figura 1

A figura 1 mostra um diagrama de precedência para um projeto simples. Ao lado de cada nó da rede, representando uma atividade, mostram-se a duração, em unidades de tempo, e as necessidades de recursos (caixa), em unidades monetárias. Aplicando-se os cálculos tradicionais do CPM, obtém-se para cada atividade um tempo cedo de início (TCI) e um tempo tarde de término (TTT). A partir destas informações, pode-se construir um diagrama de barras, com todas as atividades começando nos seus tempos cedo de início.

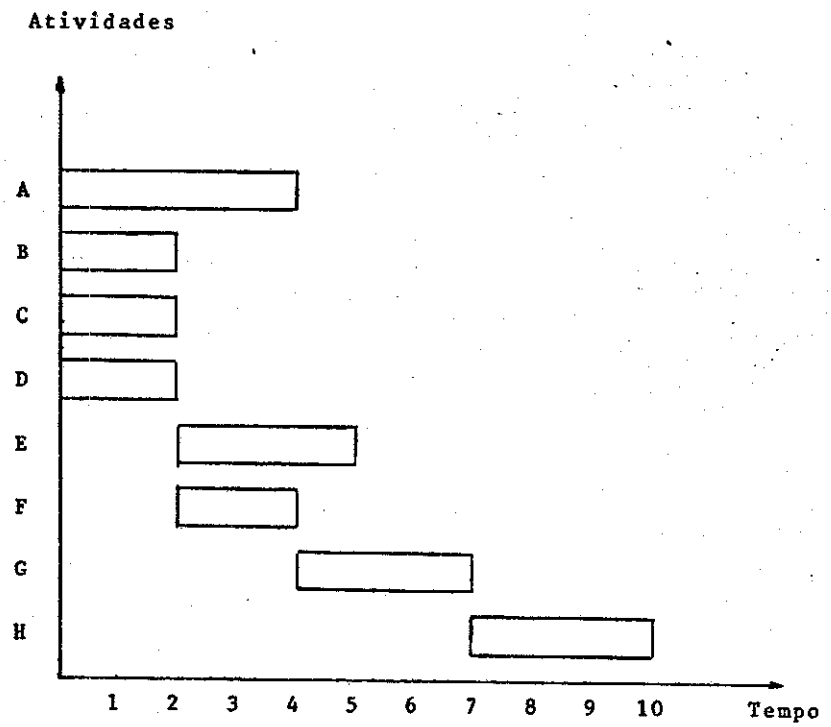


Figura 2

A partir deste gráfico de barras, constrói-se um gráfico do perfil de demanda de caixa ao longo da duração do projeto, como mostrado na Figura 3:

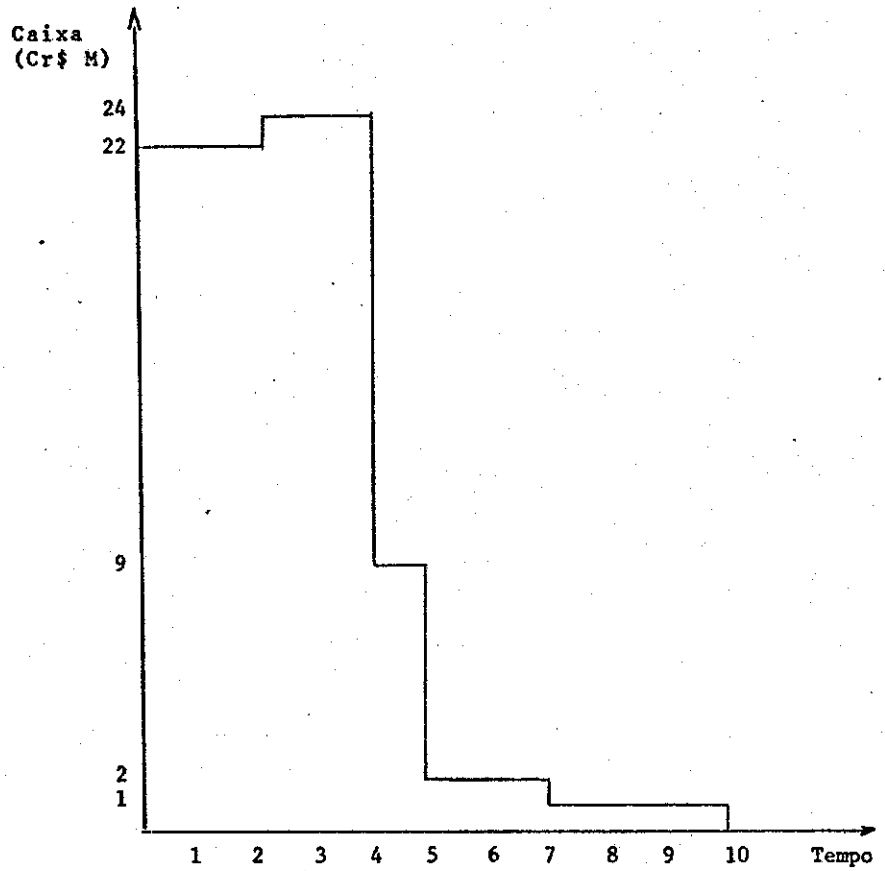


Figura 3

Neste exemplo, o perfil de utilização de recursos apresenta-se bastante elevado nos quatro primeiros meses, com uma brusca queda nos últimos seis meses do projeto, quando as necessidades de caixa tornam-se praticamente nulas em relação aos períodos anteriores.

Um típico problema de nivelamento de recursos poderia ser enunciado, informalmente, da seguinte maneira:

> O caminho crítico de uma rede já foi determinado e todas as suas atividades programadas, inicialmente, para seus tempos cedo de início. Após esta determinação, um perfil de utilização total de recursos ao longo do tempo

pode apresentar-se como na figura 3. O problema daí em diante é o de nivelar os piques de utilização dos recursos, valendo-se das folgas livres de cada atividade.

III. UMA DEFINIÇÃO FORMAL PARA O PROBLEMA

Até a presente data, inúmeros trabalhos têm sido realizados na área de resolução do problema de nivelamento de recursos em redes de planejamento. Uma pesquisa quase exaustiva da literatura, surpreendeu-nos com a inexistência de uma definição formal do problema.

Desse modo, nesta parte, pretende-se formalizar essa definição, aproveitando a possibilidade de uma rede de planejamento poder ser representada por um grafo de precedência. Assim como podemos dividir a elaboração de um projeto em um conjunto ψ de atividades elementares, com cada atividade x necessitando de um tempo t_x e de um volume de recursos $P= (\rho_1, \rho_2, \ldots, \rho_m)$ por unidade de tempo para sua realização, e como tais atividades devem também obedecer a determinadas relações de precedência, torna-se óbvia a possibilidade de representar um projeto de grande porte por um grafo de precedências.

Se a este conjunto de atividades e prioridades acrescentamos também as condições reais de escassez de recursos $\tau= (\tau_1, \tau_2, \ldots, \tau_n)$ representando a máxima disponibilidade de cada recurso ρ_i, chegamos ao problema de nivelamento de recursos em redes de planejamento, que consiste em distribuir o conjunto de atividades, no menor tempo possível, sendo obedecidas as relações de precedência e limitando as necessidades de recursos de cada atividade aos máximos disponíveis.

IV. A DEFINIÇÃO FORMAL DO PROBLEMA CLÁSSICO DE NIVELAMENTO DE RECURSOS EM REDES DE PLANEJAMENTO

Dado um conjunto finito ψ de atividades, uma relação de ordem parcial $<$ definida em ψ chamada relação tecnológica de precedência, as funções $\tau: \psi \rightarrow R^+$ (tempo de execução) e $\underline{P}: \psi \rightarrow R^+$ (necessidade de recursos por unidade de tempo) e o conjunto de constantes $T= (\tau_1, \ldots, \tau_m)$, que são as disponibilidades máximas por unidade de tempo dos m recursos necessários a cada atividade, o problema de nivelamento de recursos em redes de planejamento pode ser definido como achar uma coleção $\pi(\psi) = (S_1, S_2, \ldots, S_N)$ de subconjuntos que satisfaçam as seguintes condições:

$$\bigcup_{i=1}^{N} S_i = \psi \qquad (1)$$

garantindo que a união de todos os elementos do conjunto solução contêm todas as atividades do conjunto ψ.

$$\sum_{j \in S_k} \rho_i(y) \leq \tau_i, \quad i=1, \ldots, m \quad k = 1, \ldots, N \qquad (2)$$

garantindo que nenhuma atividade y utilize na estação S_k uma quantidade do i-ésimo recurso que seja superior à disponibilidade máxima deste recurso.

$$P(S_k) = \sum_{j \in S_k} \sum_{i=1}^{m} \rho_i(y_j) \quad K= 1, \ldots, N \qquad (3)$$

definindo que a carga total de recursos da estação S_k é igual ao somatório da utilização de todos os recursos de cada atividade y pertencentes à estação.

$$x < y \wedge x \text{ e } S_i, \; y \text{ e } S_j \rightarrow i < j \qquad (4)$$

garantindo que tarefas subseqüentes sejam alocadas a diferentes estações e que as relações tecnológicas de precedência sejam obedecidas.

$$O^* = \sum_{k=1}^{k=N} \left(\sum_{i=1}^{m} \tau_i - P(S_k) \right) \qquad (5)$$

minimizando o ócio de recursos como definido na condição acima, ou seja, que minimizemos a soma das diferenças entre o volume de recursos disponível e aquele efetivamente empregado em cada estação.

V. DEFINIÇÕES E PROPRIEDADES DO DOMÍNIO DO PROBLEMA

Para que possamos agora apresentar o algoritmo que propomos como alternativa ótima e eficiente para a solução do problema, precisamos primeiro definir a árvore de busca ou grafo do problema, assim como evidenciar algumas de suas propriedades.

V.1 - Definições

A - Estado

Um estado $X = (X_1, \ldots, X_N)$ do problema de nivelamento de recursos em redes de planejamento é uma seqüência de subconjuntos de ψ, respeitando

i) $X_i \cap X_j = \phi$, para $i \neq j \rightarrow X_i \neq \phi$, $\forall_i$

ii) $P(X_i) = \sum_{x \in X_i} \rho_x \leq \tau$, $\forall_i$

iii) $\forall\, x, y \in \psi$, $x < y \rightarrow (x \in X_i, y \in X_j, j > i)$ ou

$$Y \in (\psi - U_{i=1}^{N} X_i)$$

iv) a lista vazia é, por definição, um estado.

B - Sucessores

Um estado $Y = (Y_1, \ldots Y_M)$ é dito sucessor de um estado $X = (X_1, \ldots, X_N)$ se e somente se:

i) $M = N + 1$

ii) $Y_i = X_i$ $i \leq N$

O custo de transição C_{xy} é definido como $C_{xy} = \ell(\tau - P(Y_{N+1}))$, onde ℓ é uma função convexa crescente qualquer. Pode-se observar que $C_{xy} \geq 0$.

C - Estado Inicial (S)

É a seqüência vazia, isto é, com zero elementos.

D - Estado Terminal

Um estado é dito terminal se é uma partição de ψ. É chamado de desejado se o número de componentes dessa partição é mínimo.

E - Funções de Custo

Para cada estado $X = (X_1, \ldots, X_N)$ definimos as seguintes funções de custo:

i) $g(X)$: $X \to R^+$, que mede o custo de transição total do estado S inicial até o estado X, e é calculada somando-se o quadrado dos ócios de cada estação X_i já alocada.

ii) $h(X)$: $X \to R^+$, que avalia o custo de transição do estado X até atingirmos o estado terminal. É sempre apresentada sob a forma de seu estimador $\hat{h}(X)$.

$$\hat{h}(X) \stackrel{d}{=} (\hat{N}(X) - N) \cdot \ell\left(\theta - \frac{T^* - T(X)}{\hat{N}(X) - N}\right),$$

onde:

$$T(X) = \sum_{i=1}^{i=N} T(X_i) \quad \text{e}$$

$$\hat{N}(X) \stackrel{d}{=} N + \left\lceil \frac{T^* - T(X)}{\theta} \right\rceil^+$$

Resume-se o problema de nivelamento de recursos em redes de planejamento, então, em achar um caminho de mínimo custo entre S (estado inicial) e o conjunto de estados desejados.

V.2 - Propriedades do Grafo

Podemos observar, das definições descritas acima, que o grafo G_s tem as propriedades a seguir.

A . Propriedade 1

G_s é uma árvore anti-simétrica e, assim, existe apenas um caminho de S para qualquer estado n (donde $\hat{g}(n)=g(n)$) e, no máximo, um caminho de um estado m para outro n qualquer.

B . Propriedade 2

Se a cada estado $X = (X_1, \ldots, X_N)$ associamos N^*_X, o comprimento mínimo de uma solução passando por X, e se $Y = (Y_1, \ldots, Y_N)$, é tal que

$$\bigcup_{i=1}^{i=N} Y_i = \bigcup_{i=1}^{i=N} X_i, \; g(X) \leq g(Y)$$

Então,

$$N^*_x = N^*_y \quad f(X) \leq f(Y)$$

e dizemos que Y é dominado por X.

V.3 - O Algoritmo Proposto

A definição da árvore de buscas descrita, associada às propriedades enunciadas no item precedente para os estimadores $\hat{h}(x)$ e $\hat{N}(x)$ e aliada à utilização das condições do problema de nivelamento de recursos em redes de planejamento, origina o seguinte

algoritmo:
para cada estado $X = (X_1, \ldots, X_N)$ armazenar:

i) descrição de X
ii) g(X), N(X), T(X)
iii) $\hat{N}(X)$, $\hat{h}(X)$, $\hat{f}(X)$
iv) P(X), apontador para seu antecessor
v) F(X) = 0 (caminho aberto) ou 1 (caminho fechado)

A. Passo 1 - Inicialização

Seja s uma lista vazia; armazene s e faça

$F(s) \leftarrow 0$; $g(s) \leftarrow 0$; $N(s) \leftarrow 0$; $T(s) \leftarrow 0$; $P(s) \leftarrow \phi$

Calcule:

$$\hat{N}(s) \leftarrow \left[\frac{T^*}{\Theta} \right]^+ ; \quad \hat{h}(s) \leftarrow \hat{f}(s) \leftarrow \hat{N}(s) \cdot \ell \left(\Theta - \frac{T^*}{\hat{N}(s)} \right)$$

Faça:

$\bar{N} \leftarrow \hat{N}(s)$

B. Passo 2 - Seleção

Dentre todos os nós Y abertos, com $\hat{N}(Y) = \bar{N}$, selecione um X com $\hat{f}(X)$ mínimo, desempatando a favor de estados com $\hat{N}(X) = N(X)$ ou, arbitrariamente, caso isto não seja possível. Se não houver nós abertos, pare com erro.

C. Passo 3

$F(X) \leftarrow 1$. Se $\hat{N}(X) = N(X)$, chegou-se à solução. Caso contrário, gere todos os sucessores de X e feche X.

D. Passo 4

Para cada Y sucessor de X, calcule C_{xy}, $T(Y_{N+1})$ e faça

$N(Y) \leftarrow N(X) + 1$

$g(Y) \leftarrow g(X) + C_{xy}$

$T(Y) \leftarrow T(X) + T(Y_{N+1})$.

Calcule

$\hat{N}(Y)$, $\hat{h}(Y)$, $\hat{f}(Y) = g(Y) + \hat{h}(Y)$

$P(Y) \leftarrow X$

Se Y é dominado por algum Z aberto, elimine Y e tome o próximo sucessor de X; se isto não ocorre, e Y domina algum A aberto, elimine Z; guarde Y aberto e tome o próximo sucessor de X.

E. Passo 5

Atualize $\bar{N} = \min \{\hat{N}(X) \mid X \text{ aberto}\}$

Volte ao Passo 2.

É importante observar que a restrição da escolha, no Passo 2, a nós com $\hat{N}(X) = \bar{N}$ garante que pararemos num estado desejado (i.e., num estado terminal com número mínimo de estações). Por outro lado, sabemos que esta solução minimizará o desequilíbrio na alocação de recursos entre as estações, pois $\hat{h}(X) \leq h(X)$ $\forall X$. Mais importante que isto, a consistência de $\hat{h}$ nos permite tirar a conclusão de que o algoritmo é ótimo (no sentido de Hart[10]) para o problema, dentre todos os algoritmos admissíveis que não usem mais informações que:

i) nível máximo de recursos ζ

ii) $T^* = \sum_{x \in \psi} \zeta_x$

iii) para cada estado $X = (X_1, \ldots, X_n)$

a) N (nº de estações em X)

b) $T(X_1), \ldots, T(X_N)$

V.4 - Um Exemplo de Aplicação

A aplicação do algoritmo aqui descrito ao problema proposto por Moodie e Mandeville [7] resultou na solução que a seguir apresentamos para exemplificar a utilização do algoritmo proposto, o qual à primeira vista, parece complexo.

Repare a adaptação da rede do projeto, que mostra cada atividade dividida em n atividades de duração unitária, cada qual apresentando, no canto superior direito, o valor das necessidades de recursos por unidade de tempo.

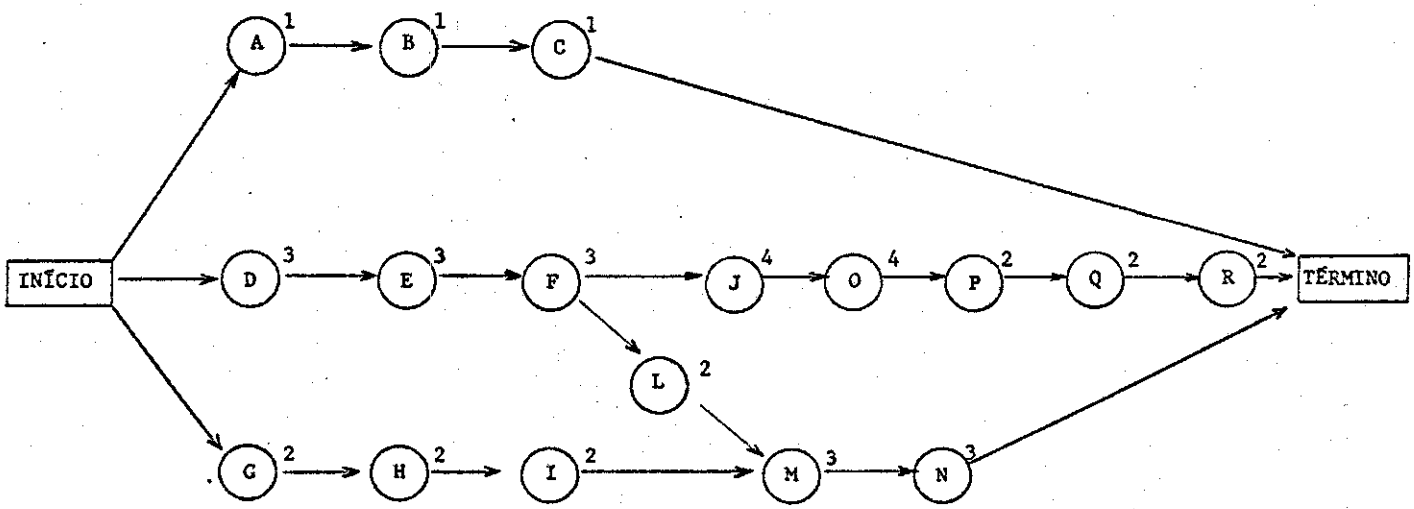


Figura 4
A rede do Projeto

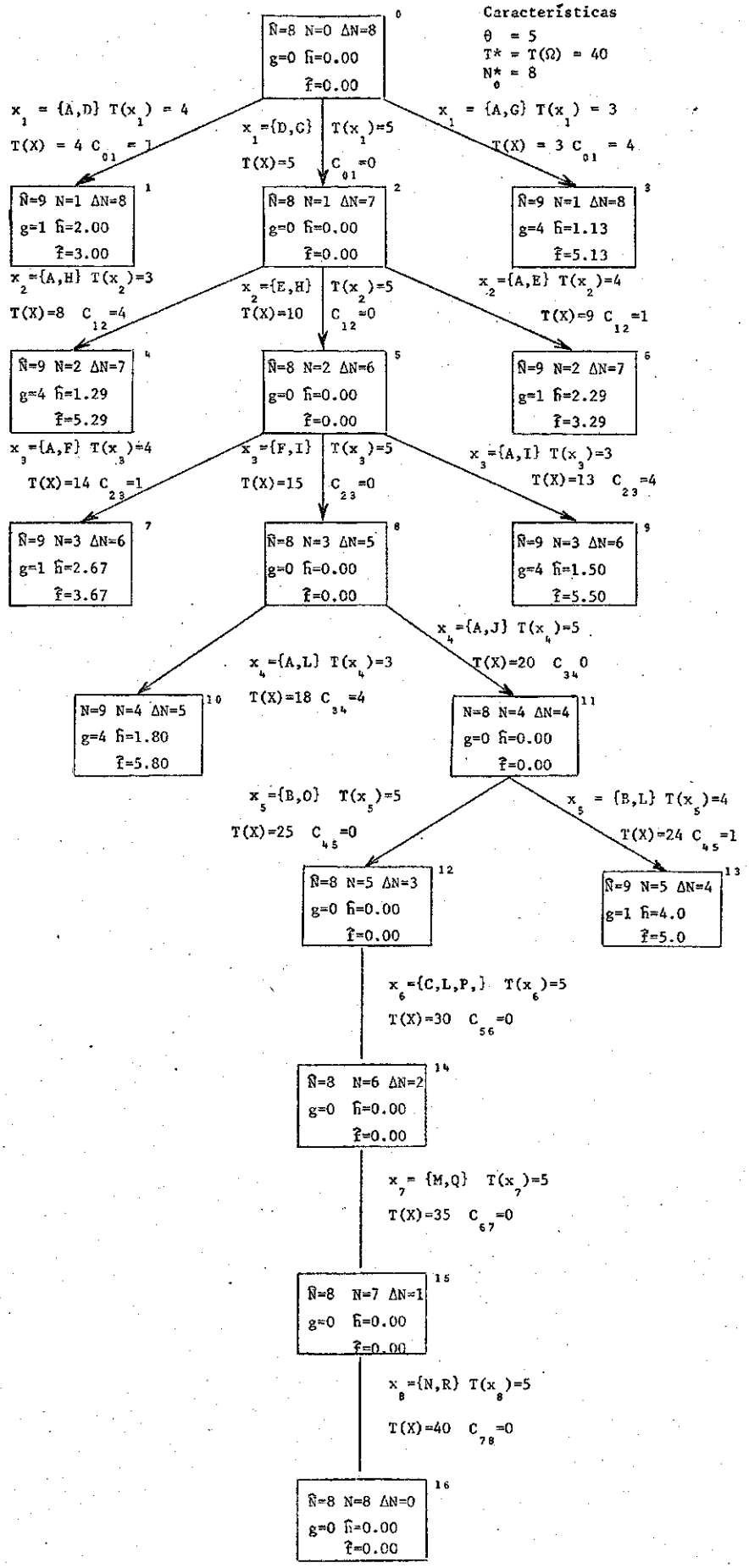
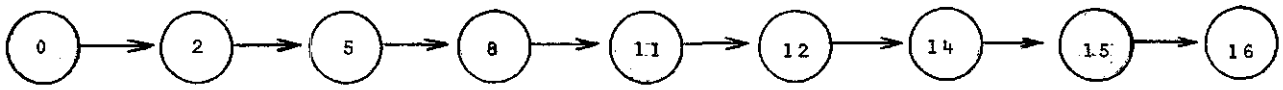


Figura 5

O Grafo do Problema

que nos leva ao caminho solução



que, por sua vez, representa o seguinte conjunto solução:

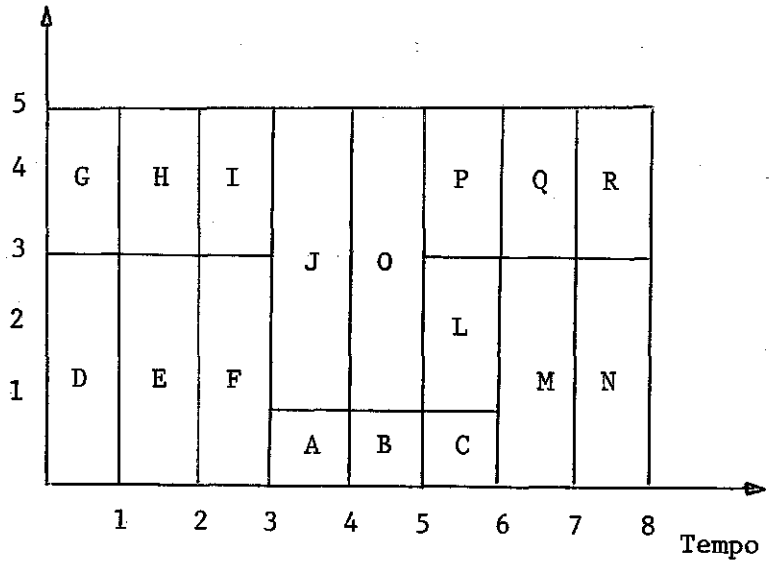
$X_1 = \{D,G\}$ $\quad$ $X_5 = \{B,D\}$

$X_2 = \{E,H\}$ $\quad$ $X_6 = \{C,L,P\}$

$X_3 = \{F,I\}$ $\quad$ $X_7 = \{M,Q\}$

$X_4 = \{A,J\}$ $\quad$ $X_8 = \{N,R\}$

cuja representação em um gráfico de alocação de recursos é:



idêntico àquele apresentado por Moodie e Mandeville através da solução por programação inteira [7].

VI. CONCLUSÃO

Acreditamos firmemente que a formalização do problema de recursos aqui apresentada ajudará no desenvolvimento de novas metodologias de ataque ao problema de alocação ótima de recursos escassos em redes de planejamento de grande porte, que, até a data presente, foi tomado como intuitivo. Mas ainda, o algoritmo aqui apresentado, como uma alternativa de solução ótima-eficiente, minimiza o desequilíbrio na alocação dos recursos às atividades sendo realizadas em paralelo no projeto, pois o algoritmo é admissível e consistente no sentido de Hart et alii [5]. Mais importante do que isto é que a consistência de $\hat{h}$ nos permite concluir que o algoritmo é ótimo dentre todos os algoritmos admissíveis que não usem mais informações do que:

i) Nível máximo de recursos ζ

ii) $T^* = \sum_{x \in \psi} \zeta_x$

iii) para cada estado $X = (X_1, X_2, \ldots, X_n)$

a) N (nº de estações em X)

b) $T(X_1), \ldots T(X_N)$

A demonstração formal das propriedades, lemas, teoremas e corolários aqui citados podem ser encontrados nas referências bibliográficas [4], [5] e [9].

BIBLIOGRAFIA

[1] BOWMANN, E.H. - Assembly line balancing by linear programming, Operations Research, Baltimore, Md, Operations Research Society of America, 8 (3) 1960.

[2] BURGESS, A. R. & KILLEBREW, J.B. Variation in activity level on a cyclic arrow diagram. Journal of Industrial Engineering, Norcross, GA, American Institute of Industrial Engineers, 13,Mar./Apr.,1962.

[3] DEWITE, L. Manpower leveling of PERT networks. Data Processing for Science/Engineering, Mar./Apr. 1964.

[4] GARCÍA GÓMEZ, Jorge A. Uma nova formulação do problema clássico de balanceamento de linhas de montagem e sua solução ótima e eficiente. Rio de Janeiro, COPPE/UFRJ, s.d., (PTS/12 79). Tese de Doutorado defendida na COPPE/UFRJ, em maio 1975.

[5] HART, P.E.; NILSSON, N.J.; RAPHAEL, B. - A Formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems Science and Cybernetics, 1968.

[6] LEVY, F.K.; THOMPSON, G.S.; WIEST, J.D. Multi-ship multi-shop workload smoothing program. Naval Research Logistics Quartely, Arlington, Wash., DC, Department of the Navy, Office of Naval Research.

[7] MOODIE, C.L. & MANDEVILLE, D.E. Project resource balancing by assembly line balancing technique. The Journal of Industrial Engineering. Norcross, GA, American Institute of Industrial Engineers, 17 (7) Jul. 1966.

[8] NEVINS, Arthur J. Assembly line balancing using best bud search. Management Science, Providence, Institute Management Sciences, 18 (9): 529-838, May, 1972.

[9] SALVESSON, M.E. The Assembly line balancing problem. The Journal of Industrial Engineering. Norcross, GA, American Institute of Industrial Engineers, 6 (3): 18-25, May/Jun. 1955.