

## PROGRAMAÇÃO MATEMÁTICA PARA CONTROLE DE TRÁFEGO URBANO

**Eduardo Camponogara**

Departamento de Automação e Sistemas

Universidade Federal de Santa Catarina

**Silvia Galvão de Souza**

Departamento de Engenharia Elétrica

Universidade Estadual de Londrina

**Werner Kraus Junior**

Departamento de Automação e Sistemas

Universidade Federal de Santa Catarina

### RESUMO

Apresenta-se uma formulação em programação matemática para o problema de controle ótimo de tráfego urbano por semáforos. As formulações são adequadas para aplicação em pacotes de otimização de problemas de Programação Linear Inteira Mista. Duas formulações são consideradas. Na primeira, propõe-se um modelo global para a rede de tráfego consistindo de nós limítrofes (entradas) e nós internos (interseções). No entanto, dada a complexidade computacional do problema global, uma formulação recursiva é proposta para problemas de interseções isoladas que pode ser empregada em conjunto com Programação Dinâmica (PD) para encontrar uma solução aproximada do problema global. O algoritmo de PD que resolve o problema de controle de uma interseção é apresentado. As duas formulações são testadas para uma rede com vias de duplo sentido e duas situações distintas para a predição de chegada de veículos: descarga de filas e chegada conhecida.

### ABSTRACT

Mathematical programming formulations for the on-line urban traffic control problem are presented. The formulations are designed to facilitate optimization by Mixed Integer Linear Programming software packages, thereby rendering the problems solvable with state-of-the-art optimization algorithms. Two formulations are considered. First, a global model for a road network consisting of boundary (entrance) and internal (intersection) nodes is presented. Then, given the computational complexity of the global problem, a recursive formulation that lends itself into a Dynamic Programming (DP) solution is derived for the single-intersection problem. A DP algorithm is also presented. Both formulations are tested for a traffic network with two-way links, considering two distinct situations for vehicle arrivals: null (queue discharge only) and known (perfect prediction) arrivals.

## 1 INTRODUÇÃO

As estratégias de controle semafórico de tráfego têm como objetivo minimizar algum critério de desempenho, tal como o número de paradas e o atraso total, através do ajuste dos tempos de verde nos semáforos. O controle de tempos fixos é, ainda hoje, a estratégia de controle mais adotada. A principal deficiência desta categoria de controle é a hipótese de que as variações do tráfego de acordo com o horário do dia e o dia da semana podem ser descritas por modelos estatísticos obtidos através de contagens de fluxo. Isto não é verdadeiro, pois a demanda pode variar tendencialmente ao longo do tempo e, também, devido a incidentes e variações de curta duração de alguns poucos ciclos.

Contrastando com tal estratégia de controle *off-line*, as estratégias *on-line* tais como a atuada pelo tráfego e adaptativa ajustam os parâmetros de controle de tráfego continuamente de acordo com as condições prevalentes do tráfego (Garbacz, 2002). A estratégia atuada pelo tráfego pode ser descrita como uma extensão da estratégia de planos de tempo fixo, pois estes planos fixos são implementados de acordo com a demanda de tráfego corrente. A estratégia de tráfego adaptativo controla grupos de tempos semafóricos para promover ganhos de desempenho em resposta às condições de tráfego passada, corrente e previstas. Em razão da flexibilidade

e substancial ganho de desempenho que pode ser obtido com controle *on-line*, esforços de pesquisa enfocam o controle através das estratégias de tráfego atuado e adaptativo.

Um exemplo de sistema de controle *on-line* é o PROLYN (Farges; Kamdem; Lesort, 1991) que faz uso de contagens veiculares obtidas através de sensores, um modelo de tráfego para prever as chegadas futuras e a técnica de horizonte deslizante para determinar o controle. O sistema PROLYN aplica programação dinâmica para resolver os problemas de otimização discreta que são instanciados pela aproximação de horizonte deslizante. Outros sistemas de controle *on-line* são OPAC (Gartner, 1983), RHODES (Mirchandani; Head, 2001), ALLONS-D (Porche et al., 1996) e CRONOS (Boilot, 1994).

As estratégias existentes de tráfego adaptativo, tais como PROLYN e OPAC, apresentam soluções para o problema de coordenação de forma implícita; tratamento explícito através de criação de níveis distintos para tratar a coordenação é apresentado em versões posteriores de OPAC (Gartner; Tarnoff; Andrews, 1991) e ALLONS-D (Porche; Lafortune, 1998). A qualidade das soluções tem motivado pesquisas para a busca de melhores métodos de coordenação, tais como modelagem de filas horizontais.

Neste artigo, um modelo de otimização global é desenvolvido, por um lado, para servir de padrão para estratégias de controle descentralizadas e, por outro lado, para auxiliar nos projetos de algoritmos de controle em tempo real sugerindo caminhos para torná-los mais efetivos e eficientes. O modelo leva em consideração a influência das ações de controle mútuas entre interseções vizinhas, a dinâmica do fluxo do tráfego e restrições ao controle. A natureza linear inteira mista do modelo (Williams, 1999) permite usar a teoria de programação inteira com a vantagem de utilização de seus algoritmos e softwares numéricos tais como Xpress-MP e ILOG Cplex que podem encontrar soluções globais ótimas para o problema de controle semafórico.

No entanto, o custo computacional de algoritmos globais pode ser proibitivo. Assim, torna-se interessante desenvolver heurísticas e algoritmos aproximativos para obter soluções práticas para cenários *on-line*. A qualidade da solução pode ser avaliada por comparação à solução obtida com o modelo global. Esta comparação pode ser feita por violação de restrições e/ou comparação de valores da função objetivo.

Neste artigo, uma heurística de solução é empregada em conjunto com um algoritmo de Programação Dinâmica (PD) para resolver de forma sub-ótima o problema de controle para uma rede de tráfego. Partindo-se de uma formulação de PD para interseção isolada, propõe-se uma abordagem iterativa como heurística para solução do problema da rede. O modelo global e o algoritmo PD serão simulados para duas situações de predição de chegadas. A primeira considera chegadas nulas, resolvendo o problema de descarga de filas com atraso mínimo, que é também utilizada no algoritmo ALLONS-D (Porche et al., 1996). A segunda considera chegadas conhecidas de veículos durante o horizonte de simulação.

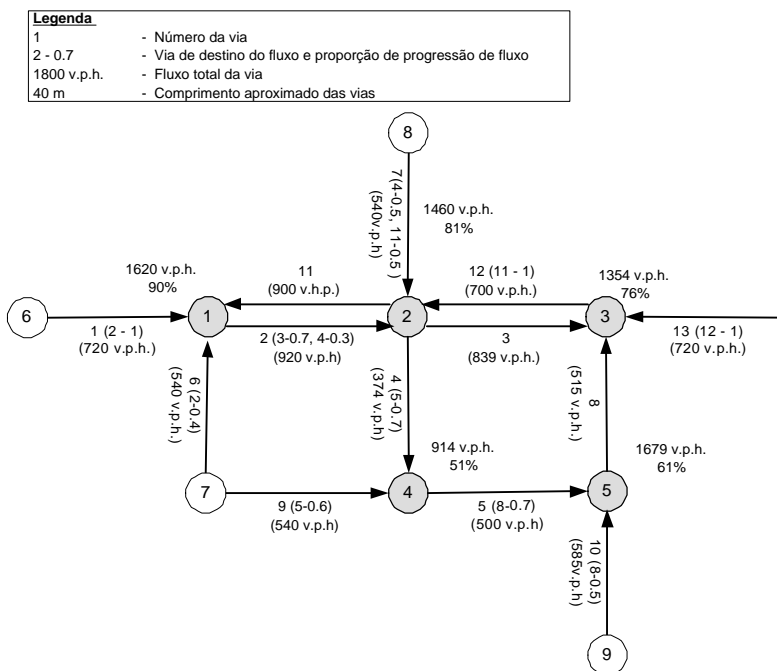
## 2 MODELO PARA PROGRAMAÇÃO MATEMÁTICA

A topologia da rede de tráfego é modelada por um grafo  $G = (V, E)$  cujos nós representam as interseções das vias e os arcos representam as vias. A existência de ciclos no grafo representa as vias com duplo sentido de fluxo. Os nós subdividem-se em dois tipos: os que representam as interseções controladas e os limítrofes, que representam semáforos sobre os quais não há autoridade de controle. Estes são estabelecidos como fronteiras da rede com o restante do sistema de tráfego. Na Figura 1, os nós-interseções, círculos escuros, são representados por

$I = \{1, \dots, 5\}$  enquanto os nós-fronteira, círculos claros, são  $B = \{6, \dots, 10\}$ , onde  $V = I \cup B$ . Os ciclos são compostos pelas vias  $L = \{2, 11\}$  e  $L = \{3, 12\}$ . A capacidade de cada via foi assumida constante e igual a 1800 v.p.h. São representados também, na Figura 1, as proporções de conversão de veículos entre as vias. A rede em consideração é denotada por  $\Gamma$ . O problema  $P$  para operação de  $\Gamma$  pode ser aproximado pelo modelo de programação matemática apresentado a seguir. O objetivo é minimizar o comprimento das filas no tempo, de acordo com a seguinte função objetivo:

$$P: \text{ Minimize } f = \sum_{k \in I} \sum_{l \in L_k} \sum_{t=0}^T x_l(t) \quad (1)$$

onde  $L_k \subseteq E$  é o subconjunto das vias que chegam na interseção  $k$ , e  $\cup_{k \in I} L_k = E$  e  $L_i \cap L_j = \emptyset, \forall i \neq j$ . Além disso,  $T$  é o número de períodos do horizonte e  $x_l(t)$  é uma variável com o número de veículos esperando na linha de parada da via  $l$  no início do período  $t$ . Note que outros critérios de desempenho podem ser utilizados, tais como atraso e número de paradas, e até mesmo combinações destes critérios.



**Figura 1:** Rede de tráfego

A dinâmica do sistema de tráfego, baseada no modelo descrito em (Farges; Kamdem; Lesort, 1991), será inicialmente apresentado aqui através de suas restrições e função objetivo (1)

formuladas para cada  $k \in I$ . O estado inicial da rede de tráfego é dado por:

Para cada  $l \in L_k$ :

$$x_l(0) = x_l^0 \quad (2)$$

$$a_{l,j}(0) = a_{l,j}^0, j = 1, \dots, N(l) + 1 \quad (3)$$

onde  $x_l^0(t)$  é o número inicial de veículos parados e  $a_{l,j}^0$  é o número inicial de veículos em velocidade livre na  $j^{\text{ésima}}$  seção da via  $l$ . O número de seções da via  $l$  é igual ao número de intervalos de controle  $t$  requeridos para atravessar o comprimento  $l$ , sendo dado por:

$$\mathcal{N}_l = \frac{L_l}{v_l t_s}, \quad (4)$$

onde  $L_l$  é o comprimento da via  $l$ ,  $v_l$  é a velocidade de cruzeiro ao longo da via e  $t_s$  é a duração do intervalo de controle. Note que  $\mathcal{N}_l$  é, em geral, não inteiro. Então, existirá um número inteiro de seções que são atravessadas em um intervalo de controle e uma seção de comprimento fracionário.

O carregamento das filas e suas descargas são formulados, para cada  $l \in L_k$  e  $t = 0, \dots, T - 1$  como:

$$x_l(t+1) \geq x_l(t) + a_{l,1}(t) - y_l(t) \quad (5)$$

$$y_l(t) \leq x_l(t) + a_{l,1}(t) \quad (6)$$

$$y_l(t) \leq S_l m_l(t) \quad (7)$$

onde  $a_{l,j}(t)$  é o número de veículos em movimento através da  $1^a$  seção da  $l^{\text{ésima}}$  via (última seção anterior à linha de parada);  $m_l(t)$  assume o valor 1 se o tráfego da via  $l$  pode fluir durante o  $t^{\text{ésima}}$  período, e 0 se não;  $S_l$  é a taxa de descarga máxima da via  $l$ ; e  $y_l(t)$  é o número de veículos que partem da via  $l$ , durante o  $t^{\text{ésima}}$  período (note que  $y_l(t) = 0$  se  $m_l(t) = 0$  e  $y_l(t) = \min\{S_l, x_l(t) + a_{l,1}(t)\}$  se  $m_l(t) = 1$ ).

A progressão ao longo das vias que levam à interseção  $k$  é dada pelas seguintes restrições:

Para cada  $l \in L_k, t = 0, \dots, T - 1, j = 1, \dots, N(l) - 1$ :

$$a_{l,j}(t+1) = a_{l,j+1}(t) \quad (8)$$

O volume de tráfego a montante da via  $l$  é dado por:

Para cada  $l \in L_k \cap W_y, t = 0, \dots, T - 1$ :

$$a_{l,N(l)}(t+1) = a_{l,N(l)+1}(t) + (1 - R_l)z_l(t) \quad (9)$$

$$a_{l,N(l)+1}(t+1) = R_l z_l(t) \quad (10)$$

$$z_l(t) = \sum_{l' \in U_l} p_{l',l} y_{l'}(t) \quad (11)$$

onde  $W_y \subseteq E$  é o subconjunto de vias que recebem fluxo de outras vias, (por exemplo,  $W_y = \{2, 3, 4, 5, 8, 11, 12\}$  na Figura 1;  $N(l) + 1$  é o número de seções da via  $l$ ,  $R_l$  é a fração inteira da via  $l$  que ocupa a  $(N(l) + 1)^{\text{ésima}}$  seção enquanto  $(1 - R_l)$  é a fração de veículos que

entram na  $N(l)^{ésima}$  seção em um intervalo;  $U_l \subseteq E$ ,  $l \in W_y$ , é o subconjunto de vias cujos veículos podem entrar na via  $l$  (por exemplo,  $U_2 = \{1, 6\}$  na Figura 1),  $z_l(t)$  é o número de veículos provenientes de outras vias que entram na via  $l$  durante o período  $t$ , vias estas correspondendo a  $l' \in U_l$ ; e  $p_{l',l}$  é a fração de veículos que trafegam da via  $l'$  para a via  $l$ . Para modelar o tráfego que entra na rede através de seus limites externos, se fazem necessárias as restrições seguintes:

Para cada  $l \in L_k \cap W_n, t = 1, \dots, T$  :

$$a_{l,N(l)}(t) = a_{l,N(l)+1}(t-1) + (1 - R_l)a_l^t \quad (12)$$

$$a_{l,N(l)+1}(t) = R_l a_l^t \quad (13)$$

onde  $W_n$  é o subconjunto de vias que recebem tráfego externo; note que  $W_y \cap W_n = \emptyset$  e  $W_y \cup W_n = E$ ; na Figura 1,  $W_n = \{1, 6, 7, 9, 10, 13\}$ .

As Equações (9-10) e (12-13) são necessárias em vista do resultado fracionário da Eq. (4).

O estado da indicação corrente, em termos do número de verdes consecutivos garantidos para a via  $l$ , iniciando no tempo  $t - 1$  e movendo-se para trás no tempo, é dado por:

Para cada  $l \in L_k, t = 0, \dots, T - 1$  :

$$q_l(t+1) \leq q_l(t) + m_l(t) \quad (14)$$

$$q_l(t+1) \geq q_l(t) + m_l(t) - M[1 - m_l(t)] \quad (15)$$

$$q_l(t+1) \leq M m_l(t) \quad (16)$$

$$g_l m_l(t+1) \geq g_l - q_l(t) - g_l M[1 - m_l(t)] \quad (17)$$

onde  $q_l(t)$  é o número de verdes consecutivos garantidos para a via  $l$  no período imediatamente precedente a  $t$ ;  $q_l^0$  é o número inicial de verdes consecutivos concedidos a via  $l$  e imediatamente anteriores ao período  $t = 0$ ;  $M$  é um número inteiro positivo suficientemente grande, onde  $M = T$  satisfaz; e  $g_l \in \mathbb{Z}_+$  é o tempo de verde mínimo expresso em períodos de controle. As condições iniciais para indicação semafórica formam um grupo de restrições adicionais:

Para cada  $l \in L_k$  :

$$q_l(0) = q_l^0 \quad (18)$$

A resolução do conflito é assegurada pelas seguintes restrições:

Para cada  $l \in L_k, t = 0, \dots, T - 1$  :

$$m_l(t) \in \{0, 1\} \quad (19)$$

Para cada  $t = 0, \dots, T - 1$  :

$$\sum_{l \in L_k} m_l(t) \leq 1 \quad (20)$$

as quais asseguram que  $m_l(t)$  assume valores binários e, para um dado tempo  $t$ , somente uma via tem permissão de passagem. A família de restrições (20) pode ser facilmente generalizada para modelar interseções que permitem múltiplos movimentos não conflitantes, ou seja, múltiplos semáforos que podem simultaneamente ter sinal “verde”.

Assume-se que todas as variáveis são números reais positivos, com exceção de  $m_l(t)$  que assume valores binários. O problema  $P$  pode ser resolvido ao encontrar a solução para (1) sujeito às restrições (2) - (20).

O problema  $P$  pode ser inserido na classe de problemas de programação linear inteira mista (Wolsey, 1998), sendo que as variáveis inteiras são as indicações semaforicas. Duas vantagens desta formulação são: (i) a obtenção limites inferiores através da solução da relaxação linear de  $P$ , dessa forma permitindo a avaliação da qualidade de soluções aproximadas ou heurísticas; (ii) o emprego de algoritmos de otimização para encontrar soluções ótimas de  $P$ , e possivelmente, soluções sub-ótimas quando um procedimento de enumeração implícita (e.g., *branch-and-bound*) sofrer uma interrupção prematura.

Em princípio, se a dinâmica do tráfego fosse determinística e se existissem recursos computacionais ilimitados, a solução de  $P$  forneceria tempos de chaveamento ótimos para todos os semáforos no horizonte de tempo  $T$ . Com relação ao primeiro impedimento (*dinâmica não determinística*), o modelo de controle preditivo instancia uma série  $\{P_t : t = 0, 1, \dots\}$  de problemas, onde cada problema  $P_t$  aproxima  $P$  sobre um horizonte limitado e faz uso de realimentação periódica para compensar os erros nas previsões, que tendem a tornar-se imprecisas à medida que o horizonte de predição é expandido. Com relação ao segundo impedimento (*recursos computacionais limitados*), sozinho, o modelo de controle preditivo não proporciona uma solução prática, assim faz-se necessário a decomposição de  $P_t$  em um conjunto de subproblemas  $\{P_{t,k} : k \in I\}$ , um para cada interseção, o que converte a série  $\{P_t\}$  em uma série de conjuntos de problemas  $\{\{P_{t,k}\}\}$ . Existem questões de natureza teórico-prática quanto à forma de decomposição de  $P_t$  em  $\{P_{t,k}\}$  e à convergência da solução distribuída de  $\{P_{t,k}\}$  para uma solução de  $P_t$ , as quais têm atraído a atenção de pesquisadores de diversas áreas.

### 3 PROGRAMAÇÃO DINÂMICA APLICADA AO PROBLEMA DA INTERSEÇÃO ISOLADA

Nesta seção, considera-se uma versão de  $P$  restrita a uma interseção isolada  $k$ , doravante referida como  $P_k$ , para a qual é desenvolvido um algoritmo de programação dinâmica (Bertsekas, 1995). São dois os benefícios deste algoritmo: em primeiro lugar, o algoritmo leva naturalmente a uma solução aproximada e eficaz para redes acíclicas (vias com mão única) e, em segundo, serve para resolver elementos de  $\{P_k : k \in I\}$  obtidos com a decomposição Lagrangeana de  $P$ .

Suponha que as variáveis  $x_l(t)$  assumem somente valores inteiros, que podem ser implementados pela declaração  $x_l(t) \in \mathbb{Z}_+$ ,  $t = 0, 1, \dots, T$  e  $\forall l \in E$ , e supondo que as constantes  $x_l(0)$ ,  $\forall l \in E$ , são todas números inteiros. Considere uma interseção  $k \in I$ . Para simplificar, seja  $a_l(t)$  a predição de chegadas de veículos no fim da fila da via  $l \in L_k$  no período  $t \in \{0, \dots, T-1\}$ , e  $m_{l'}(-1) = 1$  se a via  $l' \in L_k$  recebeu verde no período de tempo precedente, enquanto  $m_l(-1) = 0$  para todo  $l \in L_k - \{l'\}$ , e seja  $q_l(-1) \in \mathbb{Z}_+$  o número de verdes consecutivos garantidos a via  $l \in L_k$ . Na proposta de aplicação da programação dinâmica, o estado de cada estágio  $t = 0, \dots, T$  consiste de  $x^k(t) = [x_l(t), m_l(t-1), q_l(t-1) : l \in L_k]$ . As recorrências são

como segue:

$$J_T^k(x^k(T)) = \sum_{l \in L_k} x_l(T) \quad (21)$$

$$J_t^k(x^k(t)) = \text{Minimizar} \sum_{l \in L_k} x_l(t) + J_{t+1}^k(x^k(t+1)) \quad (22)$$

$$m_l(t), \forall l \in L_k$$

sujeito a :

$$\text{Seja } l^* \in L_k \text{ tal que } m^{l^*}(t-1) = 1 \quad (23)$$

Se  $q_{l^*}(t-1) < g_{l^*}$  então

$$m_l(t) = m_l(t-1), \forall l \in L_k \quad (24)$$

$$q_l(t) = q_l(t-1) + m_l(t), \forall l \in L_k \quad (25)$$

Senão

$$\sum_{l \in L_k} m_l(t) = 1 \quad (26)$$

$$m_l(t) \in \{0, 1\}, \forall l \in L_k \quad (27)$$

$$q_l(t) = \min\{g_l, q_l(t-1) + m_l(t)\}, \forall l \in L_k \quad (28)$$

Para todo  $l \in L_k$  :

$$x_l(t+1) = x_l(t) + a_l(t) - \min\{x_l(t) + a_l(t), S_l m_l(t)\} \quad (29)$$

Assumindo que o comprimento das filas é dado apenas por números inteiros entre 0 até  $n \in \mathbb{Z}_+$ , o espaço de valores viáveis para as variáveis de estado torna-se  $\Omega = \{0, \dots, n\}^{|L_k|} \times \{\alpha \in \{0, 1\}^{|L_k|} : 1^T \alpha = 1\} \times \{\beta \in \{0, 1, \dots, g\}^{|L_k|} : \beta_j \neq 0 \text{ para no máximo um } j\}$ , assumindo, para simplificar, que  $g_l = g, \forall l \in L_k$ . Então, o tamanho do espaço de estados é  $|\Omega| = \Theta(n^{|L_k|} g^{|L_k|})$ . Sob a hipótese de que os valores das variáveis são inteiros, um algoritmo de programação dinâmica pode ser concebido para resolver as recorrências acima, (21) e (22), que computa  $J_T^k(x^k(T))$  para todo  $x^k(T) \in \Omega$ ,  $J_{T-1}^k(x^k(T-1))$  para todo  $x^k(T-1) \in \Omega$ , e assim sucessivamente até encontrar o valor de  $J_0^k(x^k(0))$ .

### Algoritmo de Programação Dinâmica

Para todo  $x^k(T) \in \Omega$  faça

$$\text{Seja } J_T^k(x^k(T)) = \sum_{l \in L_k} x_l(T)$$

Para  $t = T-1$  até 0 faça

Para todo  $x^k(t) \in \Omega$  faça

$$\text{Seja } l^* \in L_k \text{ tal que } m^{l^*}(t-1) = 1$$

Se  $q_{l^*}(t-1) < g_{l^*}$  então

Para todo  $l \in L_k$  faça

$$m_l(t) = m_l(t-1)$$

$$q_l(t) = q_l(t-1) + m_l(t)$$

$$\delta_l(t) = x_l(t) + a_l(t)$$

$$\mu_l(t) = \min\{\delta_l(t), S_l m_l(t)\}$$

$$\begin{aligned}x_l(t+1) &= \min\{\delta_l(t) - \mu_l(t), n\} \\ J_t^k(x^k(t)) &= \sum_{l \in L_k} x_l(t) + J_{t+1}^k(x^k(t+1)) \\ Q_t^k(x^k(t)) &= l^*\end{aligned}$$

Senão

$$J_{min} = +\infty$$

Para todo  $l \in L_k$  faça

$$m_l(t) = 1 \text{ and } m_{l'}(t) = 0, \forall l' \in L_k - \{l\}$$

$$q_l(t) = \min\{q_l(t-1) + 1, g_l\} \text{ e } q_{l'}(t) = 0, \forall l' \in L_k - \{l\}$$

Para todo  $l' \in L_k$  faça

$$\delta_{l'}(t) = x_{l'}(t) + a_{l'}(t)$$

$$\mu_{l'}(t) = \min\{\delta_{l'}(t), S_{l'} m_{l'}(t)\}$$

$$x_{l'}(t+1) = \min\{\delta_{l'}(t) - \mu_{l'}(t), n\}$$

$$J = \sum_{l' \in L_k} x_{l'}(t) + J_{t+1}^k(x^k(t+1))$$

Se  $J < J_{min}$  então

$$J_{min} = J$$

$$l_{min} = l$$

$$J_t^k(x^k(t)) = J_{min}$$

$$Q_t^k(x^k(t)) = l_{min}$$

A tabela  $Q_t^k$  registra a ação de controle ótimo a ser tomada no tempo  $t$  para cada estado viável  $x^k(t) \in \Omega$ . Claramente, o tempo transcorrido no algoritmo de PD é  $\Theta(|\Omega| |L_k|^2 T)$ : o custo computacional corresponde a calcular todas as entradas das tabelas  $J^k = \{J_t^k : t = 0, \dots, T\}$  e  $Q^k = \{Q_t^k : t = 0, \dots, T-1\}$ , o que implica em consumo de memória da ordem  $\Theta(|\Omega| T)$ .

Em função da natureza das recorrências, o algoritmo PD encontra a solução ótima para  $P_k$  sob a condição de que as variáveis que representam o tamanho das filas sejam inteiras. Note também que o tempo de processamento do algoritmo cresce linearmente com o comprimento do horizonte e quadraticamente com o número de vias que chegam na interseção. Então, fazendo-se  $|\Omega|$  suficientemente pequeno, a aproximação de PD torna-se um algoritmo eficiente para o controle de semáforos em uma rede de interseções isoladas. Se as variáveis assumirem valores fracionários, o algoritmo de PD pode ser facilmente adaptado para resolver  $P_k$  de forma aproximada: utilize um conjunto finito de intervalos para representar de forma aproximada os valores de  $x_l(t)$ , refinando os elementos de  $\Omega$  para níveis que garantam uma solução de qualidade aceitável; por exemplo, os valores  $\{0, 0.1, 0.2, \dots, 1\}$  poderiam aproximar os valores de  $x_l(t)$  no intervalo  $[0, 1]$ .

#### 4 SOLUÇÃO HEURÍSTICA DISTRIBUÍDA DE REDES COM MÚLTIPLAS INTERSEÇÕES

Se as chegadas de cada interseção  $k \in I$  são conhecidas, se espera que a aplicação do algoritmo de PD para interseção isolada produza uma solução de boa qualidade para  $P$ . Esta observação pode levar a uma simples, mas eficaz para resolução de  $P$ :

- (1) dividir  $P$  em um grupo  $\{P_k : k \in I\}$  de problemas, um para cada interseção  $k$ , ignorando as restrições de acoplamento e assumindo que as previsões de chegadas são conhecidas;
- (2) resolver  $\{P_k\}$  em paralelo com um grupo de controladores distribuídos, usando o algoritmo de PD para cada interseção isoladamente conforme proposto na Seção 3;



- (3) atualizar as predições de chegada nas equações dinâmicas e o controle dos semáforos;
- (4) repetir a partir de (2) até que convergência seja atingida ou até que um valor máximo de iterações seja alcançado.

Esta heurística é particularmente eficaz para uma rede de tráfego representada por um grafo acíclico, isto é, redes consistindo de vias de sentido único. Neste caso, um ordenamento da sequência de solução do conjunto de problemas  $\{P_k\}$ , obedecendo uma ordenação topológica do grafo  $G$  que leva em conta os sentidos de fluxo, implica na convergência das iterações do algoritmo de PD, uma vez que o problema  $P_j$  não é afetado pelas decisões de controle referentes aos problemas  $P_{j+1}, P_{j+2}, \dots$ . Em outras palavras, a solução do problema de uma interseção a montante não é afetada pela solução de interseções a jusante. A ressalva é de que deve haver capacidade suficiente nas vias para evitar bloqueios. Tal ordenamento também está presente nas heurísticas do programa TRANSYT (Robertson, 1968) (e, portanto, no sistema SCOOT (Robertson; Bretherton, 1991)), embora não associadas à condição de aciclicidade da malha viária.

## 5 RESULTADOS

As Tabelas 1 e 2 mostram os resultados da otimização realizada para a rede apresentada na Figura 1. Todas as interseções operam com uma demanda aproximada de 70% da capacidade. Os resultados do algoritmo de programação dinâmica são comparados às soluções globais obtidas com o *software* ILOG Cplex que permite resolver problemas de programação linear inteira mista.

**Tabela 1:** Qualidade da solução distribuída de  $P \times$  solução global - descarga de filas

$T$	Programação Dinâmica	CPLEX - Global		Erro da solução PD p/solução Global
	Função Objetivo	Melhor Solução	Limite Inferior	
9	90.14	60.24	60.24	49.63 %
10	90.14	61.60	61.60	46.33 %
12	235.50	232.50	232.50	1.29 %
14	313.25	309.50	309.50	1.21 %
16	351.75	347.50	347.50	1.22 %

**Tabela 2:** Qualidade da solução distribuída de  $P \times$  solução global - chegadas conhecidas

$T$	Programação Dinâmica	CPLEX - Global		Erro solução PD s/a solução global
	Função Objetivo	Melhor Solução	Limite Inferior	
9	128.89	97.27	97.27	32.50 %
10	142.93	105.42	105.41	35.59 %
12	175.67	123.39	123.38	42.37 %
14	195.27	147.83	126.70	54.11 %
16	214.25	171.63	150.48	42.37 %

A Tabela 2 apresenta os resultados para simulações que consideram predições de chegadas conhecidas para todo o horizonte.

Já a Tabela 1 apresenta os resultados para descarga de filas, ou seja, as chegadas de veículos são conhecidas apenas no primeiro passo do horizonte, enquanto que para os passos restantes as chegadas são nulas. Desta forma é testada a descarga das filas existentes.

Para as duas situações a primeira coluna (chamada  $T$ ) mostra o número de períodos do horizonte de simulação, cada período com a duração de 4 segundos.

A segunda coluna apresenta os valores da função objetivo obtidos com a PD. Este resultado pode ser comparado com a terceira e quarta colunas que apresentam a melhor solução encontrada e o limite inferior, respectivamente, para a solução global encontrada com o CPLEX. Note que as soluções produzidas pelo *software* CPLEX são soluções globalmente ótimas pois a avaliação destas coincide com os limites inferiores.

O erro apresentado na quinta coluna é o erro do algoritmo PD comparado ao limite inferior da solução global.

## 6 DISCUSSÃO E CONCLUSÕES

O modelo e os métodos de solução para o problema de controle de tráfego urbano foram apresentados neste artigo usando o formalismo de programação matemática, permitindo a aplicação de pacotes de otimização tais como CPLEX e Xpress-MP. De acordo com os resultados apresentados, a solução ótima global pode ser encontrada para problemas de dimensão e complexidade comparáveis às instâncias que foram objeto de investigação neste artigo. Além disso, uma aproximação da solução global foi obtida por meio da combinação da Programação Dinâmica com uma heurística de sequenciamento, permitindo uma solução descentralizada de problemas em interseções isoladas. Os resultados numéricos indicam que ganhos substanciais podem ser obtidos através do aprimoramento da abordagem descentralizada. Uma alternativa é o emprego do horizonte deslizante, o que permitirá a realimentação do controle com base em predições mais acuradas e reavaliação iterativa das decisões de controle.

### Agradecimentos

Os autores agradecem ao CNPq pelo suporte à pesquisa, recebido através do financiamento ao projeto SincMobil (processo 552248/02-9), e à CAPES pela concessão de bolsa PICDT a Silvia Galvão de Souza.

### REFERÊNCIAS BIBLIOGRÁFICAS

- Bertsekas, D. P. *Dynamic Programming and Optimal Control*. [S.l.]: Athena Scientific, 1995.
- Boilot, F. Evaluation of the real time urban traffic control algorithm CRONOS: first phase. In: *Proceedings of the 7th IFAC/IFORS Symposium on Transportation*. Tianjin, China: [s.n.], 1994. p. 585–590.
- Farges, J. L.; Kamdem, I.; Lesort, J. B. *Realization and Test of a Prototype for Real Time Urban Traffic Control*. [S.l.], 1991.

- Garbacz, R. E. Adaptive signal control: what to expect. In: *Proceedings of the 12th Annual Meeting of the Intelligent Transportation Society of America*. Washington, DC.: [s.n.], 2002.
- Gartner, N. H. A demand responsive strategy for traffic signal control. *Transportation Research Record*, v. 906, p. 75–81, 1983.
- Gartner, N. H.; Tarnoff, P. J.; Andrews, C. M. Evaluation of optimized policies for adaptive control strategy. *Transportation Research Record*, v. 1324, p. 105–114, 1991.
- Mirchandani, P.; Head, L. RHODES: A real-time traffic signal control system: architecture, algorithms, and analysis. *Transportation Research – Part C*, v. 8, p. 105–114, 2001.
- Porche, I.; Lafortune, S. ALLONS-D: Coordination of local adaptive traffic signal controllers. In: *American Control Conference*. Philadelphia, PA: [s.n.], 1998. p. 585–590.
- Porche, I. et al. ALLONS-D: A decentralized scheme for real-time optimization of traffic signals. In: *Proceedings of the IEEE International Conference on Control Applications*. [S.l.: s.n.], 1996. p. 582–589.
- Robertson, D. I. *TRANSYT: A Traffic Network Study Tool*. Crowthorne, England, 1968.
- Robertson, D. I.; Bretherton, R. D. Optimization networks of traffic signals in real-time: the SCOOT method. *IEEE Transactions on Vehicular Technology*, v. 40, p. 11–15, 1991.
- Williams, H. P. *Model Building in Mathematical Programming*. [S.l.]: John Wiley, 1999.
- Wolsey, L. A. *Integer Programming*. [S.l.]: John Wiley & Sons, 1998.