

MARC KALTENBACH

Optimal control of the time-varying flows in networks

RAIRO. Recherche opérationnelle, tome 15, n° 4 (1981),
p. 335-349

http://www.numdam.org/item?id=RO_1981__15_4_335_0

© AFCET, 1981, tous droits réservés.

L'accès aux archives de la revue « RAIRO. Recherche opérationnelle » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques
<http://www.numdam.org/>

OPTIMAL CONTROL OF THE TIME-VARYING FLOWS IN NETWORKS (*)

by Marc KALTENBACH (¹)

Abstract. — Some aspects of a general problem of controlling time-varying flows in networks are addressed. Non-linear dynamical equations are obtained as a result of letting the flow accumulate at the end of the queues. It is found that an additional set of state equations offers a convenient way of enforcing upper limits on link storage capacities. For this model, having non-smooth system functions and complex (state dependent) time lags, a procedure is described for computing optimal open-loop controls. Numerical experiments document the convergence properties of the approach.

Keywords: implicit model, reduced system, gradient.

Résumé. — On considère un problème de flots dynamiques dans un réseau. Un modèle à structure logique complexe est formulé dans lequel il est tenu compte de la dynamique des queues pouvant survenir aux nœuds du réseau et des contraintes de stockage de flot sur les arcs. Pour ce modèle on décrit une procédure d'optimisation applicable à des modèles plus détaillés et réalistes. Un ensemble de résultats numériques illustre les propriétés de convergence de l'approche proposée.

Mots clés : modèle implicite, système réduit, gradient.

ACKNOWLEDGEMENTS

The author is indebted to Professor H. N. Koivo and Professor E. J. Davison for valuable comments and suggestions. Financial support by the National Research Council of Canada has been available through grants to Professors Koivo and Davison.

Major improvements in the presentation of this paper must be credited to reviewers.

INTRODUCTION

Many control problems of current interest involve systems with network structures. These problems are found in areas as diverse as the routing of data in communication networks [1, 2], water management in networks of pipes, canals and reservoirs [3], urban traffic control [4, 5], to mention only a few examples.

(*) Received September 1979.

(¹) Bishop's University Lennoxville, Quebec, Canada J1M 1Z7.

A network structure is characterized by a set of nodes and a set of directed arcs, called links, joining the nodes. Dynamics are observed when the links act as channels for the time varying flow of a commodity that may consist of either messages, fluids, vehicles, etc. Due to insufficient or conflicting service capabilities at the nodes, the flow of commodity may accumulate on the links, forming queues. A major control objective is then to manipulate the node service rates in order to optimize a function of the queues.

Analytical solutions have been obtained for problems in this class on the basis of variational calculus, [4, 5, 6]. Possibly the most advanced results to date are to be found in [2]. Due to various assumptions still seriously curbing the generality of application of these analytical results, less limited numerical solutions have also been sought as an alternative. The principal methods proposed so far have either been based on Lagrangian Theory [7], or on the Simplex Algorithm of Linear Programming [8, 9, 10].

The present paper describes an approach which is novel and yet quite in line with traditional numerical methods in Open-Loop Optimal Control [11]. Part I includes a mathematical model which is purposefully restricted to a few basic features of time varying flows in networks. One of its important originalities rests in the treatment of constraints on queue lengths. In the past, link storage capacity limits have been treated as system state constraints. This could result in system inconsistency or in optimization difficulties. In the present paper, these constraints are replaced by additional variables and dynamic equations.

The model illustrates the need in systems modelling to give a greater importance to the structure of logical relationships between systems parts. The difficulty is that the resulting system often becomes out of range of many classical optimization techniques. This problem is now widely recognized and the basis for a renewed interest in the Theory of Optimization [16].

In Part II, the paper illustrates the application of a straightforward optimization technique to the model in Part I with its non-smooth system functions and complex time lags. What is not straightforward though and should still be considered as an active area for research is how to practically implement the algorithm so as to obtain a very efficient algorithm. Due to the large amount of space required for its proper exposition, this major aspect of the work reported here is left to reference [17].

Numerical examples at the end of the paper document the quite remarkable efficiency of the algorithm obtained by the proposed method.

PART I

DESCRIPTION OF TIME VARYING FLOWS IN NETWORKS

1. A network and its elements

Consider the network in figure 1. The circles are nodes indexed in a natural way. The directed arcs, or links, are indexed relatively to the nodes they are incident to, as explained on the figure. The link indices form a set \mathcal{I} ; link index I equals zero if link I does not belong to the network. Also, let I_{-n} (respect. I_{+n}) denote the n th link upstream (respect. downstream) of link I . Link I is an entrance link if $I_{-1} = 0$, an exit link if $I_{+1} = 0$.

2. Free flow on a link

The links act as channels for the flow of some commodity assimilated to a fluid. A dynamic model is sought for the description of the distribution of commodity over the network links at successive time instants $t=0, \Delta, 2\Delta, \dots, N\Delta=T$.

At a particular time $t \in [0, T]$, a queue may cover the front part of any link $I \in \mathcal{I}$. On the remaining part, called the free flow part, the commodity, if any, travels at the constant speed V . Figure 2 depicts such a link I , at $t=n\Delta$, superimposed over a set of coordinate axis. The ordinate axis measures flow rate values, in units of commodity per second, over points of the free flow part. It is assumed that the true length of link I is sufficiently well approximated by an integral number h_I of segments called blocks; with L_I the length of link I and $V\Delta$ the length of a block, $h_I = L_I/V\Delta$. Figure 2 shows the free flow rate values to be constant over the successive blocks corresponding to intervals $(0, V\Delta]$, $(V\Delta, 2V\Delta]$, \dots , etc. The above assumptions on link lengths and flow rate values are made less restrictive by lowering the value of Δ . A queue is represented in figure 2 by a thick arrow. The dotted line over it indicates free flow rate values that have become inoperative following the inclusion of the corresponding amounts of commodity in the queue.

During a time interval Δ , the free flow distribution in figure 2 is translated by $V\Delta$ meters in the link direction. The free flow rate distribution can therefore be expressed by way of successive service rate values at node I_{-1} . Let $q_I^{\text{out}}(n)$ denote the service rate at node I in the interval $[n\Delta, (n+1)\Delta)$. If α represents the link length occupied by one unit of queueing commodity, then the queue is progressing at speed $\alpha q_I^{\text{out}}(n)$. This implies that a queue behaves as an incompressible fluid since changes of speed occur simultaneously over the whole

length of the queue. Requirement that queue speeds be less than V leads to the general condition on all free flow rate values q ,

$$0 \leq q < \frac{V}{\alpha}. \quad (1)$$

3. Queue dynamics

Let $x_I(n)$ denote the amount of commodity queueing on link I at time $n\Delta$. Let $\Delta q_I^{\text{in}}(n)$ represent the amount of commodity joining queue x_I during time interval $[n\Delta, (n+1)\Delta)$. Flow conservation implies,

$$x_I(n+1) = x_I(n) + \Delta [q_I^{\text{in}}(n) - q_I^{\text{out}}(n)]. \quad (2)$$

(a) Description of $q_I^{\text{in}}(n)$

A mathematical expression is sought for $q_I^{\text{in}}(n)$ when the flow joins the queue by its left end in figure 2. This leads to a new expression for (2).

At time $n\Delta$, the number of blocks entirely covered by queue x_I is:

$$v_I(n) \stackrel{\Delta}{=} \text{Int} \left[\frac{\alpha x_I(n)}{V\Delta} \right],$$

where $\text{Int}[z]$ denotes the integral part of the real number z ⁽²⁾.

The free flow rate values on link I are, in shortened notation:

$$q_i \stackrel{\Delta}{=} q_{I-1}^{\text{out}}(n+i-1-r_I); \quad i=1, 2, \dots, r_I; \quad r_I \stackrel{\Delta}{=} h_I - v_I(n), \quad (3)$$

q_1 corresponds to the block in which x_I terminates. The part of this block covered by the queue is $V\delta$, where:

$$\delta \stackrel{\Delta}{=} \frac{\alpha}{V} x_I(n) - \Delta v_I(n), \quad (4)$$

δ is expressed in seconds and its meaning follows from the fact that δ/Δ is the fraction of the block covered by a queue.

⁽²⁾ Sign $\stackrel{\Delta}{=}$ denotes an equality defining the symbol on its left.

A question to answer is where to place the end of queue x_I at $t=(n+1)\Delta$ in figure 3 representing the resulting distribution of commodity over link I . The method of solution is recursive. Suppose this end has not been found in the blocks, indexed $i=1, \dots, k-1$, corresponding to the q_i values at $t=n\Delta$, translated by $V\Delta$ meters on the right in figure 3; assume the end is in block k and define δ_k by specifying δ_k/Δ to be the fraction of block k covered by a queue.

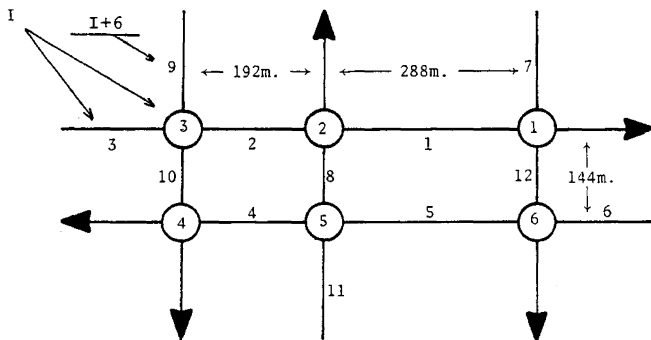


Figure 1. - A network with 6 nodes.

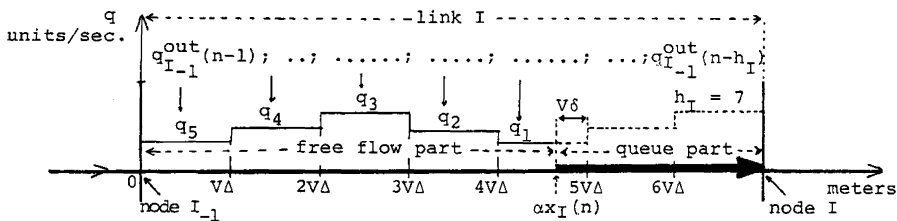


Figure 2. - Flow distribution on link I at time $n\Delta$.

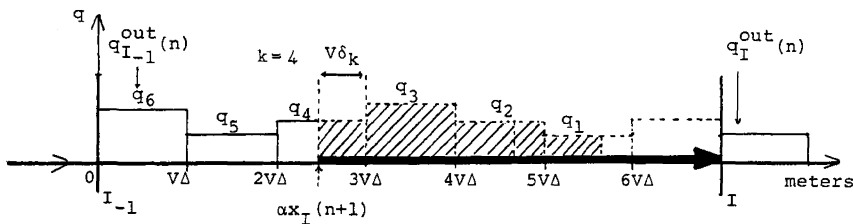


Figure 3. - Flow distribution on link I at time $(n+1)\Delta$.

Now, (i) equate queue lengths with corresponding link lengths, i. e.,

$$[x_I(n+1) - x_I(n)]\alpha = [\delta_k + (k-2)\Delta - \delta]V, \quad (5)$$

(ii) write that $\Delta q_I^{\text{in}}(n)$ equals the amount of commodity over the parts (shaded in figure 3) of blocks 1, 2, ..., k that overlap with queue x_I at $t = (n+1)\Delta$, i. e.,

$$\Delta q_I^{\text{in}}(n) = \delta_k q_k - \delta q_1 + \Delta \sum_{i=1}^{k-1} q_i \quad \text{with} \quad \sum_{i=1}^0 (\cdot) \stackrel{\Delta}{=} 0. \quad (6)$$

From equations (2), (5), (6) obtain an expression for δ_k which, substituted in (5), gives,

$$x_I(n+1) = x_I(n) + \frac{V\Delta}{V - \alpha q_k} \left[\delta(q_k - q_1) - (k-2)q_k - q_I^{\text{out}}(n) + \sum_{i=1}^{k-1} q_i \right] \quad (7)$$

a valid expression only if x_I ends in block k , i. e., if:

$$[x_I(n+1) - x_I(n)]\alpha < [(k-1)\Delta - \delta]V. \quad (8)$$

From (7), (8) derive the relevant k as the smallest positive integer satisfying,

$$\left(\frac{\alpha}{V} - q_1 \right) \delta - q_I^{\text{out}}(n) + \sum_{i=1}^k q_i < (k-1) \frac{\alpha}{V}. \quad (9)$$

Observe that (1) implies $k \geq 1$ (and, if $k=1$, $\delta_1 > \delta$); conversely a value k exceeding $r_j + 1$ would violate a condition insured by ulterior model specification (15). Hence the search for k is limited to $1 \leq k \leq r_j + 1$. That search does not constitute a computational burden in a simulation. It can be shown that for realistic simulation figures, the k values obtained are small, most often $k=1$. On the other hand it seems there is no need to select a high h_j value ($r_j < h_j$) in order to obtain a good link length approximation.

(b) *Determination of $q_I^{\text{out}}(n)$*

Denote by $S_I u_I(n)$, $n=0, 1, \dots, N-1$, the maximum flow rate out of link I into link I_{+1} ; S_I is a scaling parameter and $u_I(n)$ is the control action during time interval $[n\Delta, (n+1)\Delta)$. In this model, the control action at a node is analogous to that of a valve [2, 4] in the sense that if I and I' are two links incident to node I then increasing the service rate capacity for one direction means decreasing the service rate capacity for the other. This complementary is expressed by the condition:

$$u_I(n) + u_{I'}(n) = 1. \quad (10)$$

Furthermore, in order to prevent service to be monopolized by one direction, introduce the bounds,

$$0 < U_{\min} \leq u_I(n) \leq U_{\max} < 1. \tag{11}$$

In this model the actual flow rate out of node I will normally be

$$q_I^{\text{out}}(n) = S_I u_I(n), \quad n = 0, 1, \dots, N-1, \tag{12}$$

where, in view of (1),

$$S_I < \frac{V}{\alpha U_{\max}}.$$

However there may occur two cases in which the control action $u_I(n)$ ceases to be effective and (12) does not hold.

(i) there may not be sufficient demand to fill the service capacity provided by (12). Then, to prevent $x_I(n+1) < 0$, impose:

$$q_I^{\text{out}}(n) \leq \frac{(\Delta - \delta) q_I + x_I(n)}{\Delta}; \tag{13}$$

(ii) due to storage capacity limitations, link I_{+1} may not be able to admit all the flow from link I .

As an aside, let C_J denote the maximum storage capacity, in units of commodity, allowed on link $J \in \mathcal{J}$; then from Section 2, $C_J \leq L_J/\alpha$. Define also $\Delta G_J(n)$ as the total amount of commodity on link J at $t = n\Delta$. Then $G_J(n)$ satisfies the conservation equation:

$$G_J(n+1) = G_J(n) + q_{J-1}^{\text{out}}(n) - q_J^{\text{out}}(n). \tag{14}$$

Returning to the main argument, it is required that $\Delta G_{I_{+1}}(n+1) \leq C_{I_{+1}}$, which combined with (14) for $J = I_{+1}$, gives:

$$q_I^{\text{out}}(n) \leq \frac{C_{I_{+1}}}{\Delta} - G_{I_{+1}}(n) + q_{I_{+1}}^{\text{out}}(n). \tag{15}$$

Finally, summarize (10), (13) and (15) in:

$$q_I^{\text{out}}(n) = \text{Min} \left[\frac{(\Delta - \delta) q_I + x_I(n)}{\Delta}; \quad S_I u_I(n); \quad \frac{C_{I_{+1}}}{\Delta} - G_{I_{+1}}(n) + q_{I_{+1}}^{\text{out}}(n) \right]. \tag{16}$$

This completes the description of the dynamics associated with a generic link of the network. Only slight modifications are needed to cover the case of entrance and exit links; e. g., if $I_{+1} = 0$, omit (15) in (16) (infinite sink assumption).

4. System simulation

Form the system equations by collecting pairs of equations (7), (14) for all networks links. Provide initial conditions $x_I(0)$, q_i , $i = 1, 2, \dots, r_I$, $\forall I \in \mathcal{I}$ and, $\forall I \in \mathcal{I}$ such that $I_{-1} \neq 0$, compute,

$$G_I(0) = \frac{(\Delta - \delta)q_1 + x_I(0)}{\Delta} + \sum_{i=2}^{r_I} q_i.$$

Select a control trajectory from the admissible set:

$$\mathcal{U} = \{ U(n) | n = 0, \dots, N-1; \text{ Vector } U(n) \text{ components}$$

are the independent control actions at the network nodes,

$$\text{and satisfy (11)} \} \quad (17)$$

Iterate the system equations for N time steps. Observe that within each time step the individual equations cannot be iterated in arbitrary order since (16) implies that $q_I^{\text{out}}(n)$ must be computed after $q_{I_{+1}}^{\text{out}}(n)$ if $I_{+1} \neq 0$. This apparent obstacle to the possibility of further extending the model to networks allowing flow circulation over cyclic paths may be overcome by deleting $q_{I_{+1}}^{\text{out}}(n)$ from the Min argument in (16). This achieves link decoupling at the price of underestimating link I_{+1} capacity by:

$$\Delta \left(\frac{V}{\alpha} - q_{I_{+1}}^{\text{out}}(n) \right),$$

a small quantity with Δ .

PART II

OPTIMAL CONTROL

A common measure of performance is the sum over a prescribed control period $[0, T]$ of all delays incurred by units of commodity for reason of insufficient service capacities at network nodes [2, 4]. This total delay is most simply expressed [19] by setting $\alpha = 0$ in the model just described and computing,

$$PI = \Delta \sum_{n=0}^N \sum_{I \in \mathcal{I}} x_I(n); \quad T = N \Delta. \quad (18)$$

Equation (7) then reverts to the fixed delay equation,

$$x_l(n+1) = x_l(n) + \Delta [q_{l-1}^{\text{out}}(n-h_l) - q_l^{\text{out}}(n)]. \quad (19)$$

However equation (16) remains unchanged in form. On the other hand, the model with $\alpha > 0$ can be used to give a closer approximation [19] to total kinetic energy loss, an important index of performance in urban vehicular traffic networks [5]. In this paper the major interest is not on the realism of the model, but rather on optimization techniques for systems with complex structural features. Accordingly optimization of the proposed model with $\alpha > 0$ shall be considered for (18) and for:

$$PI = \sum_{n=0}^N X'(n) Q X(n), \quad (20)$$

where vector $X(n)$ has components $x_l(n)$, $l \in \mathcal{I}$ and Q is a square matrix. This quadratic PI may be viewed as a way of penalizing long queues.

1. Statement of an optimal control problem

A numerical strategy using successive approximations is sought for the open loop optimal control problem:

Find a control schedule in \mathcal{U} (17), which minimizes PI in (18) or (20), subject to a network flow system of generic dynamic equations (7), (14) with given (i. e. fixed) initial conditions.

This problem is non-trivial as it involves non-smooth system functions and complex time lags. On the other hand, the system representation is unconventional in the sense that the complete form of the system difference equations for a given network is never explicitly stated. Since the relevant parts of the system equations are generated only when required in the course of a specific simulation or optimization, the model is said to be implicit. Explicit equations could be written, but they would involve such a large number of terms that it would be difficult to put them to practical use. On the other hand the complexity of an explicit representation would seriously limit the computational efficiency of an optimization algorithm applied to it. At present, the author is not aware of an alternative optimization approach to the one described in the following. It consists in the application of one of several possible gradient approaches [11] to the system model in implicit form.

2. Formal descent algorithm

Obtain reduced forms for the system equations by assuming k fixed in (7) and equating $q_i^{\text{out}}(n)$ to the active argument of the Min operator in (16). Record this information in an index $IDX(I, n)$. Then array IDX , of components $IDX(I, n)$, $I \in \mathcal{I}$, $n = 0, \dots, N-1$, characterizes the sequence of reduced system functions associated with a particular system simulation. With respect to the entire set \mathcal{U} , there is a finite number p of different vectors IDX . Partition \mathcal{U} with respect to the equivalence relation which associates two elements in \mathcal{U} if they result in the same IDX . Obtain the resulting family of connected subsets $\mathcal{P} = \{\mathcal{S}_i | i = 1, \dots, p\}$ where \mathcal{S}_i is the subset corresponding to a particular value of IDX denoted IDX_i . If the nominal control trajectory, denoted μ_l with l referring to a particular gradient iteration and $\mu_l \in \mathcal{U}$, lies in the interior \mathcal{S}_i of set \mathcal{S}_i , then within some neighborhood of μ_l the system functions can be replaced by their reduced forms which are then differentiable throughout the neighborhood. Any one of the classical gradient algorithms with well studied convergence properties may be applied to this reduced system [11]. Now assume μ_l belongs to the common boundary $\partial \mathcal{S}_{ij} \stackrel{\Delta}{=} \partial \mathcal{S}_i \cap \partial \mathcal{S}_j$ of two sets \mathcal{S}_i and \mathcal{S}_j . Since PI is continuous with respect to $\mu \in U$, associating either IDX_i or IDX_j to μ_l does not change the value of PI . Observe that the reduced system functions corresponding to IDX_i (respt. IDX_j) are directionally differentiable with respect to all directions in \mathcal{U} pointing toward \mathcal{S}_i (respt. \mathcal{S}_j). Let g_i denote the gradient vector $\partial PI / \partial \mu |_{\mu = \mu_l}$ computed assuming $IDX = IDX_i$. If g_i points toward \mathcal{S}_i , then the situation is the same as if $\mu_l \in \mathcal{S}_i$. If none of g_i and g_j have the above property, a direction of descent is sought in $\partial \mathcal{S}_{ij}$, or within a neighborhood of that set, by projecting g_i and g_j on $\partial \mathcal{S}_{ij}$. For a local optimum at $\mu = \mu_l$ it is necessary that both g_i and g_j be orthogonal to $\partial \mathcal{S}_{ij}$. This reasoning extends to cases where μ_l belongs to the common boundaries of more than two sets in \mathcal{P} .

This formal approach reduces the given problem to a succession of subproblems for which well founded methods of solution exist. Observe however that the sets \mathcal{S}_i are very small and that computed directions of descent are most likely to be valid over large groups of contiguous sets in \mathcal{P} . This has motivated experiments with the following simpler, accelerated descent algorithm.

3. Relaxed structure gradient algorithm

Step 0 : Set $l=0$; select $\mu_l = [U(0), \dots, U(N-1)]$; determine the corresponding IDX value by iterating the system equations for $n=0, 1, \dots, N-1$.

Step 1 : Compute the gradient $\partial PI/\partial \mu_l$ corresponding to *IDX*. This may be done with the help of the algorithm in [12]. Note that fixing *IDX* (for this step only) results in differentiable reduced system functions.

Step 2 : Perform a step size search along a direction derived from the gradient. Set $l=l+1$; record the best *PI* with corresponding μ_l and *IDX* values. The controls are kept in \mathcal{U} by the procedure indicated in [13].

Step 3 : If a sufficient improvement of *PI* has been obtained at step 2, go to step 1, if not stop.

The convergence of this simplified procedure has been studied theoretically for the case $\alpha=0$ (storage at the nodes). It can be shown that the linear performance index (18) defines a convex function over the closed convex set of admissible controls. Precisely, the objective function surface is obtained as $\text{Max} \{H_1, H_2, \dots, H_m\}$ where the H_i 's are hyperplanes in $R^{II \times N+1}$, and *II* denotes the number of links in the network. Consequently any non optimal solution admits a direction of descent and any local minimum is a global minimum. The simple version of the algorithm presented here does not rule out a possibility of "jamming" along an edge formed by two of the hyperplanes just mentioned. However anti-jamming procedures may be added to the algorithm. One procedure tested consists in combining descent directions derived from system structures identified at successive descent steps, in accordance with the conjugate gradient algorithm [11]. This procedure is applicable only when few structure changes are involved in a descent step; which is the case when the algorithm approaches jamming conditions. Note that if many structure changes were involved in a descent step, the gradient information from past descent steps would lose significance and only contribute to slowing and possibly prevent convergence. This fact has been observed experimentally. Another anti-jamming procedure [16] would consist in including in the gradient computation those $u_l(n)$ variables which are within a small ε range of defining an active reduced system structure.




The gradient descent method just presented bears a close analogy to the *GRG* approach [14] in that the gradient is computed with respect to a local system structure (the set of active constraints) and the modification of the decision variables along the negative gradient direction may extend beyond the strict range of validity of the local system structure.

In the *GRG* approach the necessity of maintaining all variables in a feasible domain [13] and explicitly solving some implicit functions in order to update the local system structure, is often the source of much computational complexity that is avoided in the present approach. In counterpart it will not always be possible to lump external constraints in the system functions as done in the

present paper, and when possible, a difficult gradient computation may result. Thus it appears that in some cases there will be a definite advantage to combine the two approaches.

4. Numerical examples

For the network in figure 1, let $V=12$ m/sec.; $\alpha=4$ m; $\Delta=4$ seconds; $N=98$, $S_I=2.75$ for $I=1, \dots, 12$; $U_{\min}=1/3$; $U_{\max}=2/3$. The system initial conditions are all zero, except for some non-zero flow rates distributed periodically with period 14Δ upstream on entrance links. Namely, if $d_i, i=1, 2, \dots$, is a periodic distribution with first period $[.6, 2, 2.5, 2, 1.5, .8, .3, 0, 0, 0, 0, 0, 0]$, then for links 3, 7 and 11, $q_i = \text{Min}[1.3 d_i, 2.5]$ and for links 6 and 9, $q_i=0$ if $i \leq 7$ and $q_i = \text{Min}[1.3 d_{i-7}, 2.5]$ if $i > 7$. A first objective will be to minimize PI in (20), with Q a (12×12) unit matrix. Several computer runs are started with different initial nominal control trajectories, denoted CTRL(l) $l=1, \dots, 4$. For $l=1, 2, 3$, CTRL(l) corresponds to constant control actions, respectively: $u_I(n) = U_{\min}$; $u_I(n) = .5$ and $u_I(n) = U_{\max}$, for $I=1, \dots, 6$ and $n=0, \dots, N-1$. CTRL (4) corresponds to the crude feedback law: $u_I(n) = U_{\max}$ if $x_I(n) > x_I(n)$ and $u_I(n) = U_{\min}$ otherwise; I' denotes the vertical link incident to node I . Table 1 displays the PI values, corresponding to the four initial control sequences, before and after optimization. A good agreement is obtained in the final PI values. The slight variations in final values are attributed to the gradient algorithm termination and step size selection procedures as well as numerical round off errors. A set of optimized control trajectories is represented by the thick lines in figure 4. Observe that these trajectories rapidly fall into periodic patterns; this no doubt reflects the periodic arrivals specified at the network entrances. A proper phasing of the control actions at adjacent nodes may also be observed. In this model, the optimal controls are not necessarily bang-bang. Bang-bang controls were obtained in [6] for the case of cumulated arrivals at the network entrances and removal of queue size constraints. In the present case, it is easy to formulate a counter example for a one intersection network.

For the flow out of link I , the controls $u_I(n)$ are ineffective when $q_I^{\text{out}}(n) < S_I u_I(n)$. Accordingly, in figure 4, the areas shaded  (respect. ) denote ranges of values $u_I(n)$ which have no effect on the flow out of horizontal link I (respect. vertical link I' incident to node I). Consequently $u_I(n)$ values are arbitrary in the range of areas shaded .

In the above examples, convergence has been quite fast. All runs took less than 30 seconds cpu time on an IBM 370/168 computer operating under MVS (VS 2). The fact that in all cases 80% of the cost reductions were obtained within the first 4 seconds of a run is significant because operational applications are unlikely to

require such a degree of convergence as reported in table 1; all intermediate solutions are system feasible.

TABLE 1

Optimization results

Initial control sequences	Corresponding quadratic (linear) PI values	Optimized quadratic and Corr. lin. PI values	Optimized linear PI values
CTRL (1) [U_{min}]	105 493 (18 600)	54 166 (15 240)	14 416
CTRL (2) [.5]	103 051 (18 452)	54 101 (15 228)	14 480
CTRL (3) [U_{max}]	136 961 (22 816)	54 165 (15 244)	14 424
CTRL (4) [Feedback].	95 769 (21 343)	54 102 (15 224)	14 404

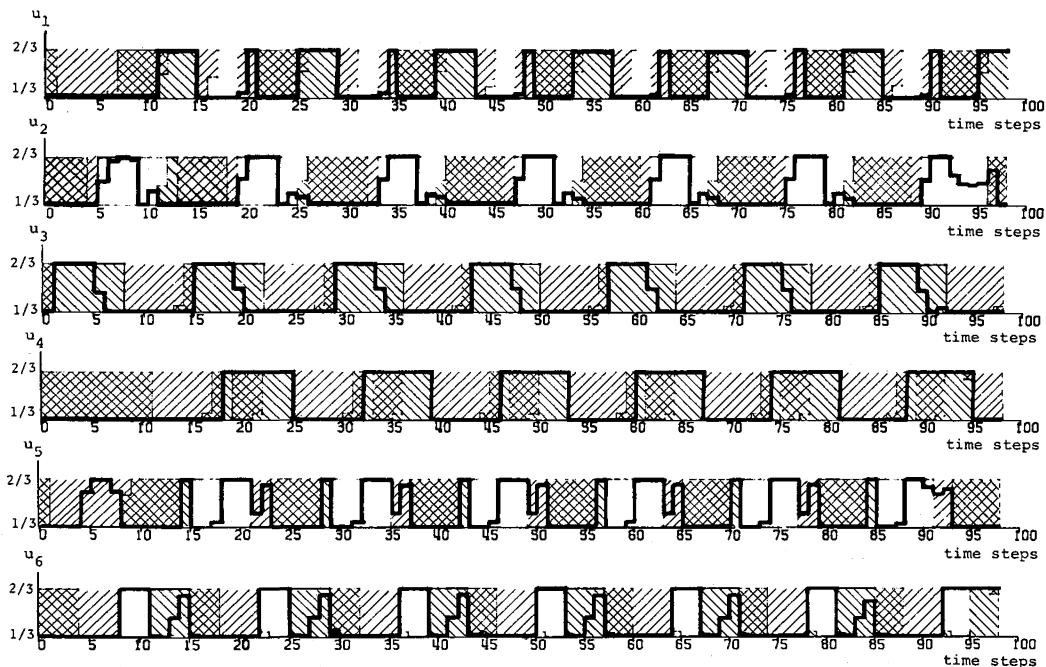


Figure 4. — Optimized control trajectories

Problems of numerical accuracy may restrict this approach to networks not exceeding 13-17 nodes depending on the network topology, the selected PI , the number of time steps and the flow intensity.

CONCLUSION

This paper has reported the successful application of a gradient type optimization technique to a model of quite complex logical structure.

In summary, the basic idea of the method is that of first looking for ways of eliminating external constraints on the model dynamic equations and then applying an optimization technique based on consideration of only the local (or active) structure of the system equations.

The interest of this approach is that it can be applied to more complex systems which are the rule, rather than the exception, in reality. For instance, the network flow model can be extended [18] to the more realistic case of compressible flows; the model obtained keeps track of various queue segments of different densities. This causes a sharp increase in the number of logical conditions to be checked at each stage of the system simulation as well as in the number of possible local system structures.

At present, models of that type are expected to fall beyond the range of applicability of purely analytical methods of optimization. Even when theoretically possible, the analytical solutions may require consideration of too many particular cases to be practical. The present approach eliminates this diversity. However, its proper implementation still represents a major task. At present the proper exploitation of special structure poses challenges at least equal to that of developing general optimization techniques in earlier times.

There are direct applications for the small network flow optimization algorithm described in this paper. On the other hand the algorithm can be used as one component of a (sub-optimal) feedback loop for the real time decentralized control of flows in very large networks [17]. One major idea in that scheme is to consider overlapping subnetworks optimized by parallel computations. Coordination between the subnetworks is achieved naturally by the overlapping of the subnetworks. More specific details are contained in the author's thesis.

REFERENCES

1. A. SEGALL, *Dynamic File Assignment in a Computer Network*, I.E.E.E. Trans. Aut. Contr., Vol. AC-21, April 1976, pp. 161-173.
2. F. H. MOSS, *The Application of Optimal Control Theory to Dynamic Routing in Data Communication Networks*, Ph.-D. thesis, M.I.T., February 1977.
3. M. JAMSHIDI and W. HEIDARI, *Application of Dynamic Programming to Control Khuzestan Water Resources System*, Automatica, Vol. 13, No. 3, May 1977, pp. 287-293.
4. D. C. GAZIS, Ed., *Traffic science*, Wiley, 1974.
5. K. K. BURHARDT, *Urban Traffic System Optimization*, Ph.-D. Thesis, Univ. of Minnesota, 1971.
6. M. KALTENBACH and H. N. Koivo, *Modelling and Control of Urban Traffic Flow*, Proc. J.A.C.C., Austin, Texas, 1974, pp. 147-154.
7. M. G. SINGH and H. TAMURA, *Modelling and Hierarchical Optimization for Oversaturated Urban Road Traffic Networks*, Int. J. Contr., Vol. 20, 1974, pp. 913-934.
8. G. C. D'ANS and D. C. GAZIS, *Optimal Control of Over-Saturated Store-And-Forward Transportation Networks*, Transp. Sci., Vol. 10, February 1976.
9. H. TAMURA, *Multistage Linear Programming for Discrete Optimal Control with Distributed Lags*, Automatica, Vol. 13, No. 4, July 1977, pp. 369-376.
10. A. P. WIERZBICKI, *Reduced Gradient in Multistage Linear Programming*, Automatica, Vol. 13, No. 4, July 1977, pp. 441-442.
11. E. POLAK, *Computational Methods in Optimization*, Acad. Press, 1971.
12. S. E. DREYFUS, *The Computational Solution of Optimum Control Problems with Time Lags*, I.E.E.E. Trans. Aut. Contr., Vol. AC-18, No. 4, August 1973, pp. 383-385.
13. V. H. QUINTANA, and E. J. DAVISON, *Clipping-Off Gradient Algorithms to Compute Optimal Controls with Constrained Magnitude*, Int. J. Contr., Vol. 20, 1974, pp. 243-255.
14. J. ABADIE, *Application of the GRG Algorithm to Optimal Control Problems*, chapt. 8 in *Integer and Non linear Programming*, J. ABADIE, Ed., North-Holland Pub. Cy., Amsterdam, 1970.
15. H. T. BANKS and A. MANITIUS, *Application of Abstract Variational Theory to Hereditary Systems — a Survey*, I.E.E.E. Trans. Aut. Contr., Vol. AC-19, October 1974, pp. 524-533.
16. C. LEMARECHAL, *Nonsmooth Optimization: Toward a Synthesis*, Abstracts, Optimization, Days, Ecole Polytechnique, Montreal, May 1978.
17. M. KALTENBACH, *Modelling and Control of Traffic Networks*, Ph.-D. Thesis, U. of Toronto, 1976.
18. M. KALTENBACH, *Analysis of Queue Dynamics in Area Traffic Network*, Research Report, No. 1, 1979.
19. M. KALTENBACH, *Delay and Kinetic Energy Losses in Dynamic Queues*, Research Report, No. 2, 1979.