PROCEEDINGS

IEEE INTERNATIONAL CONFERENCE ON CIRCUITS AND COMPUTERS

ICCC 80

EDITED BY
N. B. GUY RABBAT

Volume 1 of 2

OCTOBER 1-3, 1980
RYE TOWN HILTON INN
PORT CHESTER, NEW YORK

IEEE CATALOG NO. 80CH1511-5
LIBRARY OF CONGRESS
CATALOG CARD NO. 79-90696

BOUNDED FLOW EQUILIBRIUM PROBLEMS BY PENALTY METHODS

D. W. Hearn
and
J. Ribera

Department of Industrial and Systems Engineering
University of Florida
303 Weil Hall
Gainesville, Florida 32611

ABSTRACT

In this paper we consider the convex cost multicommodity network flow problem often used in traffic assignment models and the packet switching problem, but with the additional restriction that there be bounds on the individual arcs of the network. The addition of such bounds raises both conceptual and computational questions which have been addressed by Hearn [5], Payne and Thompson [11], Daganzo [1] and Hearn and Ribera [6]. We briefly review the results of these papers and then introduce a new penalty Lagrangian method patterned after the techniques of Pierre and Lowe [12] and Rockafellar [13]. This penalty approach is chosen because it is applicable to the very large scale networks employed in practical models. The results of some computational tests are also presented.

1. INTRODUCTION

The problem we consider is

$$(P) \quad \min_x \sum_a f_a(x_a)$$

$$\text{s.t.} \quad Bx^k = b_k \qquad k \in D$$

$$x^k \geq 0$$

$$\ell_a \leq x_a \leq u_a \qquad \text{for all } a$$

where

x_a^k = flow to the $k^{\underline{th}}$ destination on link a

x^k = vector of flows on all links to destination k

$x_a = \sum_k x_a^k$ = total flow on link a

B = arc-node incidence matrix for the network

b_k = demand vector for destination k. Entries are positive for origins, negative for destination k, and zero for intermediate nodes.

D = set of destination nodes

ℓ_a, u_a = lower and upper bounds on total link flows $(0 \leq \ell_a < u_a)$

$f_a(x_a) = \int_0^{x_a} t_a(z)\, dz$

$t_a(x_a)$ = per unit cost (time) of link a as a function of total flow

a = subscript of a typical link joining node i to node j.

We assume throughout that all $t_a(\cdot)$ are nonnegative, convex, nondecreasing functions. Note that if $\ell_a = 0$ and $u_a = \infty$ for all a, then (P) is a well-known [2, 3, 9, 10] nonlinear programming problem, the optimal solution of which is a set of flows satisfying Wardrop's "user-equilibrium principle." According to this principle, the cost of every utilized path between an origin and destination is a constant no greater than the cost of any nonutilized path. Of course the cost of a path is the sum of link costs for links in the path. This principle has appeal to transportation planners and to the designers of packet switching networks, so (P) is a much studied model for which many solution methods have been proposed [2 - 4, 8 - 10]. An alternative to Wardrop's principle is the so-called "system-optimization principle" for which the objective is minimization of total cost to all users. To find a set of flows obeying this principle, one would alter (P) to have $\sum_a t_a(x_a)x_a$ as an objective and, again $\ell_a = 0$ and $u_a = \infty$. The computational results of this paper are applicable in a straightforward manner to the system-optimization model, so, for brevity, we consider just (P) as stated. Also, it should be noted that adding bounds $(0 < \ell_a < u_a < \infty)$ presents no conceptual problem for the system-optimization model - that is, the optimal solution would simply minimize total cost subject to the bounds.

The theorem of the next section interprets the solution of (P) when the bounds are present. Then in the following sections we address computational questions.

2. CONSTRAINED EQUILIBRIA

The following theorem and corollary are proven by Hearn [5]. They generalize and extend an

CH1511-5/80/0000-0162 $00.75 © 1980 IEEE

earlier result due to Payne and Thompson [11].

Theorem (Constrained User Equilibrium Principle)

Let α_a and β_a be the (nonnegative) Karush-Kuhn-Tucker multipliers for problem (P) associated with optimal flows x_a^*. Then on any utilized path joining an origin to destination k the total path cost = $\sum_{\text{path}} t_a(x_a^*) - \alpha_a + \beta_a$ is constant and less than the same total cost on any nonutilized path. The cost on arcs for which $\ell_a < x_a^* < u_a$ is simply $t_a(x_a^*)$, i.e., $\alpha_a = \beta_a = 0$.

Corollary

Consider an equilibrium problem (E) which is the same as (P) except that (i) the constraints $\ell_a \leq x_a \leq u_a$ are deleted, and (ii) the link cost functions are altered to be $t_a(x_a) - \alpha_a + \beta_a$. Then the flows x_a^* are optimal for (P) only if they are optimal for (E). If the x_a^* uniquely solve (P), then a solution of (E) also solves (P).

As a simple example, consider the trivial network of Figure 1.

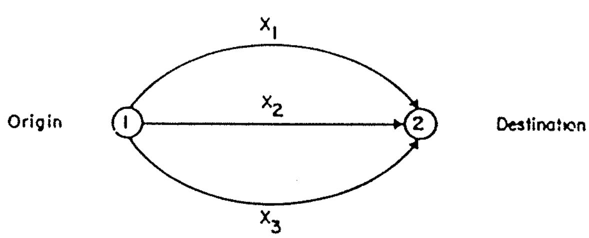

FIG. 1

Let the link costs be $t_a(x_a) = (x_a)^2$, a = 1, 2, 3. Then (P) is

$$\text{(P1)} \quad \min \frac{x_1^3}{3} + \frac{x_2^3}{3} + \frac{x_3^3}{3}$$

$$x_1 + x_2 + x_3 = 12$$

$$0 \leq x_1 \leq 2 \qquad x_2 \geq 6 \qquad x_3 \geq 0$$

which has as a solution $x_1^* = 2$, $x_2^* = 6$, $x_3^* = 4$. If the bounding constraints were not present, the solution would be $x_1^* = x_2^* = x_3^* = 4$ and the cost on each utilized path(=link) would be 16, i.e., in equilibrium. Clearly this property does not hold for the optimal solution to (P1).

The nonzero multipliers for (P1) are $\beta_1 = 12$ and $\alpha_2 = 20$. From the theorem, the total path cost, including these terms, is 16 on each path. Employing the corollary, we have problem (E1), below, which is a normal equilibrium model - the link costs have been altered to $t_1(x_1) = (x_1)^2 + 12$, $t_2(x_2) = (x_2)^2 - 20$, and $t_3(x_3)$ is unchanged. The bounds do not appear, but the solution is the same.

$$\text{(E1)} \quad \min \quad \frac{x_1^3}{3} + 12x_1 + \frac{x_2^3}{3} - 20x_2 + \frac{x_3^3}{3}$$

$$\text{s.t.} \quad x_1 + x_2 + x_3 = 12$$

$$x_1, \quad x_2, \quad x_3 \geq 0$$

For transportation planners, one practical use of the above results is correction of the user-equilibrium model data. It often happens that a given network model, with thousands of links, will produce link flows which do not obey bounds known to the experienced planner. He is then forced to correct the data, especially the link cost formulas, until the user-equilibrium solution is "within bounds". From the above results we see that the multipliers of (P) can guide this correction process. For further discussion and other applications, see [5].

3. COMPUTATIONAL CONSIDERATIONS

Almost any nonlinear programming algorithm will solve (P) in theory. However, in many practical applications, the network is so large that storage and handling of the data required by the algorithm becomes the limiting factor. It is not unusual for an urban street network model to consist of 10,000 links or more. For this reason, the Frank-Wolfe [9] algorithm has become the primary solution method used in practice, despite the fact that there are many algorithms with better convergence properties. This algorithm is a feasible direction method with the unique feature that the direction finding step, when applied to (P), decomposes into a set (one for each origin) of minimum spanning tree problems if the bounding constraints are not present. Further, storage of just the total flows (the x_a, but not the x_a^k) as well as the $t_a(x_a)$ is all that is required.

When the bounding constraints are present, the Frank-Wolfe method subproblems are no longer minimum spanning tree problems (See Klessig [8]) and the resulting computational complexities make the Frank-Wolfe (or any other) method prohibitive.

For these reasons we have sought to solve (P) by penalty methods. In [6] Hearn has introduced the penalized problem

$$\text{(Q)} \quad \min \quad \sum_a f_a(x_a) = r\, P_a^1(x_a) + r\, P_a^2(x_a)$$

$$\text{s.t.} \quad Bx^k = b_k$$

$$x^k \geq 0$$

where $P_a^1 = \frac{1}{m_1} \bar{t}_a \left(\max\left(0, x_a - u_a\right)\right)^{m_1}$

$$P_a^2 = \frac{1}{m_2} \bar{t}_a \left(\max\left(0, \ell_a - x_a\right)\right)^{m_2}$$

$\bar{t}_a$ = constant

$m_1 = m_2$ = integer constants

r = positive parameter

Note that if $x_a > u_a$ or $\ell_a > x_a$ then either $P_a^1 > 0$ or $P_a^2 > 0$, but for $\ell_a \leq x_a \leq u_a$, both penalty terms are zero. Problem (Q) can be solved by the standard Frank-Wolfe algorithm and it can be proven that as $r \to \infty$, the sequence of solutions, $x_a(r)$ of (Q) tend to some x_a^* which solves (P). Some computational results on this are in [5], along with convergence proofs. A drawback to the approach arises in obtaining numerical values for the α_a and β_a. Defining $\tilde{\beta}_a(r) = r \frac{dP_a^1}{dx_a}(x_a(r))$, it can be proven that $\tilde{\beta}_a(r) \to \beta_a$ as $r \to \infty$. Thus β_a is the limit of two quantities, one tending to infinity and the other to zero. (Of course, α_a exists as a similar limit.) This numerical instability leads to rather poor estimates of β_a and α_a, hence we have attempted the penalty Lagrangian approach of the next section. For this approach, the numerical stability problems do not exist (in theory).

It is worth mentioning a very special case of (P) for which a variation of the Frank-Wolfe method does work, which has been introduced by Daganzo [1]. Specifically, he assumes $\ell_a = 0$ for all a and further that $\lim_{x_a \to u_a} f(x_a) = \infty$ for all a. Under these assumptions it is clear that $x_a^* < u_a$ for all a, i.e., the bounds are redundant to the problem. In this case Daganzo points out that truncation of the line searches in the Frank-Wolfe method (so that no bound is violated) will yield a convergent method. Actually, the assumption that $\lim_{x_a \to u_a} f_a(x_a) = \infty$ is overly restrictive, and we have shown in [6] that Daganzo's modified method will work on a much broader class of problems for which $x_a^* < u_a$.

4. COMPUTATIONAL RESULTS

Two augmented penalty Lagrangean functions patterned after the original ideas of Powell [14] and Hestenes [7], but modified to deal with inequality constraints have been tested on sample problems. Namely, the penalty Lagrangean functions of Rockafellar [13],

$$L_w(x, \beta) = \sum_a f_a(x_a) + P_a(x_a, \beta_a)$$

where

$$P_a(x_a, \beta_a) = \begin{cases} w(x_a - u_a)^2 + \beta_a(x_a - u_a) & \text{if } x_a - u_a \geq -\beta_a/2w \\ -\beta_a^2/4w & \text{otherwise} \end{cases}$$

and that of Pierre and Lowe [12]

$$L_w(x, \beta) = \sum_a f_a(x_a) + P_a(x_a, \beta_a)$$

where

$$P_a(x_a, \beta_a) = \begin{cases} w(x_a - u_a)^2 + \beta_a(x_a - u_a) & \text{if } \beta_a > 0 \\ w(x_a - u_a)^2 & \text{if } \beta_a = 0 \text{ and } x_a > u_a \\ 0 & \text{otherwise} \end{cases}$$

have been applied to two sample networks. Problem 1 corresponds to Figure 1 above with data summarized in Table 1. Results corresponding to problem 1 under a variety of penalty factors are displayed in Table 3 for the Rockafellar function and in Table 4 for the Pierre and Lowe function. (The multiplier update formulas are given in [13] and [12].) As a performance measure used to evaluate the convergence of the computed values to the optimal values, d has been defined as the maximum relative error in x and β at the end of an iteration (multiplier update).

The final rows in Tables 3 and 4 represent a heuristic mixed penalty strategy (w_0, w_1, Δw) consisting of a large penalty w_0 updated (i.e., reduced by a factor Δw) together with the multipliers until the lower bound w_1 is reached.

Problem 2 is an 18 link network (see Figure 2 and Table 2) from [5] which had previously been solved using (Q). Results with the Pierre and Lowe function are summarized in Table 5. Here 10 Frank-Wolfe iterations were allowed and the mixed strategy (5, 1, 1.1) was employed. As with problem 1, this strategy gives more satisfactory convergence results, especially when compared to those obtained using (Q).

ACKNOWLEDGEMENTS

This work was supported in part by NSF Grant ENS-7925065 and by the U. S. Army Research Grant DAAG29-80-C-0050 through the Center for Mathematical System Theory, University of Florida, Gainesville, FL 32611.

REFERENCES

[1] Daganzo, C. F., "On the Traffic Assignment Problem with Flow Dependent Cost I and II," Transportation Research, 11, pp. 433-441, (1977).

[2] Dafermos, S.C., "An Extended Traffic Assignment Model with Applications to Two-Way Traffic", Transportation Science, 5, pp. 366-89, (1971).

[3] ________, and F. T. Sparrow, "The Traffic Assignment Problem for a General Network", J. Res. National Bureau of Standards-B, 73B, (April 1969).

[4] Fratta, L., M. Gerla, and L. Kleinrock, "The Flow Deviation Method: An Approach to Store-and-Forward Communication Network Design", Networks, 3, 97-133 (1973).

[5] Hearn, D. W., "Bounding Flows in Traffic Assignment Models, "TR 80-4, ISE Department, University of Florida, (1980).

[6] Hearn, D. W. and J. Ribera, "On Daganzo's Frank-Wolfe Method for Certain Bounded Variable Traffic Assignment Problems," TR 80-3, ISE Department, University of Florida, (1980).

[7] Hestenes, M. R., "Multiplier and Gradient Methods," Computing Methods in Optimization Problems - 2, Academic Press (1969).

[8] Klessig, R. J., "An Algorithm for Nonlinear Multicommodity Flow Problems," Networks, 4, pp. 343-355, (1974).

[9] LeBlanc, L. J. Morlok, E. K., and Pierskalla, W. P., "An Accurate and Efficient Approach to Equilibrium Traffic Assignment on Congested Networks," Transportation Research Record 491, Interactive Graphics and Transportation Systems Planning, pp. 12-33 (1974).

[10] Nguyen, S., "An Algorithm for the Traffic Assignment Problem," Transportation Science, 8, pp. 203-16, (1974).

[11] Payne, H. J. and W. A. Thompson, "Traffic Assignment on Transportation Networks with Capacity Constraints and Queuing," Presented at 47th ORSA/TIMS meeting, April 30, 1975.

[12] Pierre, D. A. and M. J. Lowe, Mathematical Programming via Argumented Lagrangians, Addison-Wesley, (1975).

[13] Rockafellar, R. T., "The Multiplier Method of Hestenes and Powell Applied to Convex Programming" JOTA, Vol. 12, No. 6, (1973).

[14] Powell, M. J. D., "A Method for Nonlinear Constraints in Minimization Problems," Optimization, Academic Press (1972).

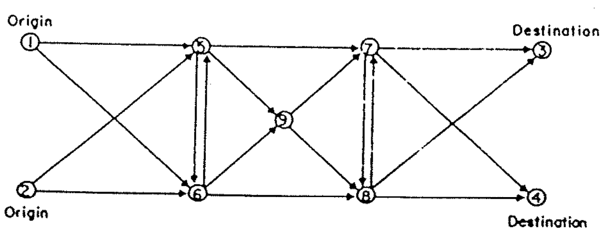

FIG. 2

$$f(x) = \sum_{a=1}^{3} \frac{c_a}{2} x_a^2 \qquad \text{s.t.} \qquad x_1 + x_2 + x_3 = 10$$

$$0 \le x_a \le u_a$$

		SOLUTION	
$c_1 = 60$	$u_1 = 5$	$x_1 = 4$	$\beta_1 = 0$
$c_2 = 50$	$u_2 = 3$	$x_2 = 3$	$\beta_2 = 90$
$c_3 = 30$	$u_3 = 3$	$x_3 = 3$	$\beta_3 = 150$

Table 1. Data for Problem 1

$$f(x) = \sum_{a=1}^{18} c_a x_a (1 + .03\,(x_a/b_a)^4) \qquad 0 \le x_a \le u_a$$

Link	c_a	$b_a = u_a$
1-5	5	12
1-6	6	18
2-5	3	35
2-6	9	35
5-6	9	20
5-7	2	11
5-9	8	26
6-5	4	11
6-8	6	33
6-9	7	32
7-3	3	25
7-4	6	24
7-8	2	19
8-3	8	39
8-4	6	43
8-7	4	36
9-7	4	26
9-8	8	30

Trip Table

	3	4
1	10	20
2	30	40

Table 2. Data for Problem 2

Penalty W	d				First iterate where $d \leq 0.01$
	It. 5	It. 10	It. 25	It.200	
100	20.20	31.90	21.340	Unstable	——
25	42.97	11.53	0.120	Unstable	——
10	92.46	51.32	8.840	0.003	81
5	123.80	90.79	36.660	0.001	158
400, 10, 1.2	43.29	3.78	0.075	0.004	38

$$d = \max \left[\|x - x^*\| / \|x^*\| \ , \ \|\beta - \beta^*\| / \|\beta^*\| \right]$$

Table 3. Computational Results for Problem 1 Using Rockafellar Function

Penalty W	d				First iterate where $d \leq 0.01$
	It.5	It.10	It.25	It.200	
100	2.89	0.05	0.012	0.008	15
25	42.77	11.05	0.187	0.003	36
10	92.44	51.30	8.830	0.003	81
5	123.80	90.79	36.660	0.001	158
400, 10, 1.2	2.44	0.43	0.004	0.001	22

Table 4. Computational Results for Problem 1 Using Pierre-Lowe Function

Link	Capacity	It.5		It.10		It.25		Prior Solution [5]	
		X	β	X	β	X	β	X	β
1-5	12	12.13	0.00	11.87	1.59	12.00	0.03	11.81	0.00
1-6	18	17.86	8.04	18.12	6.44	18.00	7.95	18.19	7.82
2-5	35	35.12	1.05	34.94	2.01	34.98	0.78	34.87	0.00
2-6	35	34.87	2.76	35.05	1.78	35.01	2.94	35.13	2.57
5-6	20	10.79	0.00	9.57	0.00	9.97	0.00	9.17	0.00
5-7	11	10.66	32.18	11.05	28.90	11.01	32.52	11.34	26.73
5-9	26	25.78	7.65	26.18	4.59	26.00	7.69	26.18	8.40
6-5	11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6-8	33	33.17	15.74	32.93	17.50	33.00	14.86	33.20	9.60
6-9	32	30.36	0.00	29.81	0.00	30.00	0.00	29.29	0.00
7-3	25	25.08	3.09	24.90	3.87	24.93	0.56	25.17	2.51
7-4	24	17.43	0.00	17.31	0.00	16.98	0.00	19.86	0.00
7-8	19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8-3	39	14.90	0.00	15.09	0.00	15.06	0.00	14.83	0.00
8-4	43	42.56	0.00	42.67	0.00	43.02	3.28	40.14	0.00
8-7	36	5.82	0.00	5.16	0.00	4.91	0.00	7.49	0.00
9-7	26	26.03	11.82	26.00	12.46	25.99	13.33	26.19	6.53
9-8	30	30.12	5.88	30.00	6.73	30.00	4.76	29.27	0.00

Table 4. Computational Results for Problem 2