An Analysis of Natural Approaches for Solving Multycommodity-Flow Problems

Alberto Caprara[1], Antonio Frangioni[2], Tiziano Parriani[1]

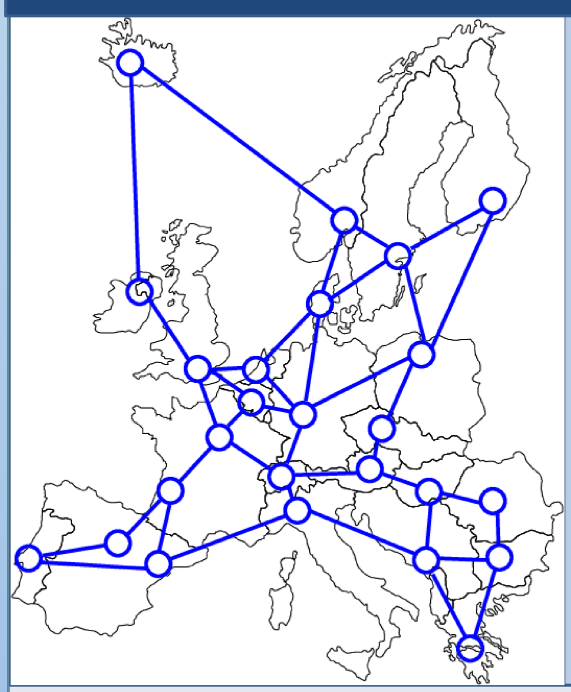
1 DEIS, University of Bologna, Viale Risorgimento 2, I-40136, Bologna, Italy

2 DI, University of Pisa, Largo B. Pontecorvo 3, I-56127 Pisa, Italy

INTRODUCTION

The linear MultiCommodity Flow problems (MCF) have naturally attracted the interest of researchers in operations research, first as basic optimization models for network design and operation problems, and then as a natural illustration of decomposable structures in linear optimization.

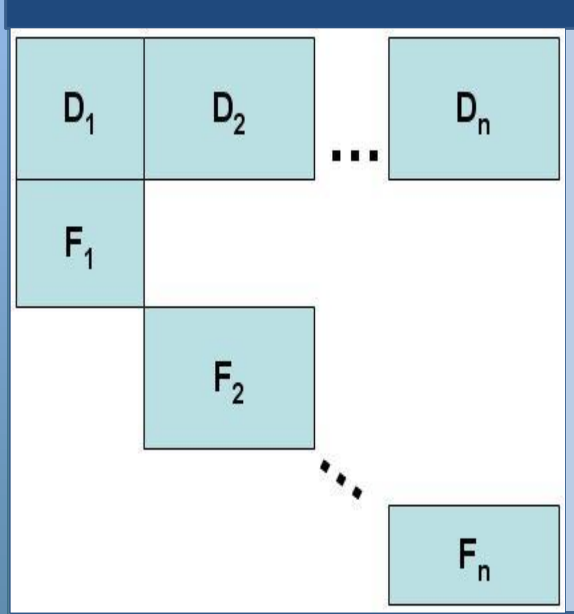
The Multicommodity Flow Problem



The MCF is characterized by a set of commodities to be routed through a network at a minimum cost. Typically related to telecommunication and transportation systems, MCFP arises in a wide variety of important applications.

In many real-world problems, the "straightforward" approach, i.e. to solve the LP corresponding to the natural node-arc formulation with general-purpose commercial software, used to require unacceptable solving times. Thus, in the last decades different approaches have been developed for solving MCF. Most of these approaches try to take advantage of the block diagonal structure typical of the constraint matrix of MCF. Among these approaches, one of the most successful is the Dantzig Wolfe decomposition (DW).

The Dantzig Wolfe Decomposition



The DW is a general framework that takes advantage of the block diagonal structure of the constraint matrix of a linear program. DW decomposes the problem in a finite number of "easy" subproblems, corresponding to blocks in the constraint matrix. The idea is to reformulate the model in terms of the extreme points (and rays) of the "easy" subproblems and then applying a column generation technique, exploiting the fact that solving a large number of "easy" subproblems may be faster than solving the "difficult" original problem.

Thanks to commercial software enhancement, in recent years the performance, in terms of solution time, obtained using the above-mentioned straightforward approach has become comparable, and in many cases better, than that of the "standard" DW approach. Nevertheless, it has been shown that the performances of the DW technique can be significantly improved by using devices able to cope with the most common limitation of column generation.

OBJECTIVES

The purpose of this work is to better understand the connections between the structure of a specific MCF instance and the performances of the most common solving approaches for this class of problems.

MOTIVATIONS

Characterizing the impact of the structure of a MCF instance onto the performances of the different available solving approaches will help in choosing the correct approach based on the features of the particular MCF instance that shall be solved. Furthermore, it may help in improving the effectiveness of the approaches themselves.

METHODS

We study three existing approaches to solve the minimum cost linear MultiCommodity Flow Problem (MCFP). The first (a.1) is solving the LP corresponding to the natural node-arc formulation (1)-(5) with state-of-the-art, general-purpose commercial software, which implement different variants of the simplex algorithm (possibly exploiting network techniques e.g. to construct a crash basis) and/or interior-point techniques

$$\min \sum_{h \in K} \sum_{(i,j) \in A} c^h_{ij} x^h_{ij} \qquad (1)$$

$$\sum_{j:(i,j) \in A} x^h_{ij} - \sum_{j:(i,j) \in A} x^h_{ji} = b^h_i \quad \forall i \in N, \forall h \in K \qquad (2)$$

$$0 \leq x^h_{ij} \leq u^h_{ij} \quad \forall (i,j) \in A, \forall h \in K \qquad (3)$$

$$\sum_{h \in k} x^h_{ij} \leq u_{ij} \quad \forall (i,j) \in A \qquad (4)$$

$$x^h_{ij} \geq 0 \quad \forall (i,j) \in A, \forall h \in K \qquad (5)$$

The second (a.2) is to develop Dantzig-Wolfe decomposition/column generation approaches to solve the path reformulation, (6) – (10).

$$\min \sum_{p \in P} c_p x_p \qquad (6)$$

$$\sum_{p \in P_h} x_p = b^h_{s_h} \quad \forall h \in K \qquad (7)$$

$$\sum_{h \in K} \sum_{p \in P_h : (i,j) \in p} x_p \leq u_{ij} \quad \forall (i,j) \in A \qquad (8)$$

$$\sum_{p \in P_h : (i,j) \in p} x_p \leq u^h_{ij} \quad \forall (i,j) \in A, h \in K \qquad (9)$$

$$x_p \geq 0 \quad \forall p \in P \qquad (10)$$

The third (a.3) is the decomposition-based pricing procedure, proposed by Mamer and McBride in [8], in which the same subproblems of the D-W decomposition are used to identify new columns in a reduced master problem that has the same structure of the node-arc formulation, thereby being in some sense "in between" a.1 and a.2 .

RESULTS

The relative performances of these approach vary significantly on different classes of problems. A small sample of this behavior is shown in Table 1 for three sets of instances arising in different application contexts. The first two sets are well known MCF instances. The second set is composed of graph formulations of real world Train Timetabling Problem (TTP) instances [2]. For the three sets of instances at hand, a.2 and a.3 overall behave notably better than a.1 on the TTP instances, as it is the case for many other real-world applications, while on the PDS and MNET instances a.1 is by far the best method.

	a.1	a.2					a.3				
Inst.	Time	Number of			Time		Number of			Time	
		Rows	Col.s	Iter.s	Master	Col.Gen.	Rows	Col.s	Iter.s	Master	Col.Gen.
MNET-256-16-4	0.45	9890	7131	464	165.22	6.4	14484	9651	314	73.45	4.19
MNET-256-64-4	2.76	37327	27685	469	2557.87	26.6	55327	38087	327	860.08	15.83
PDS-18	4.93	7291	1712	1692	63.13	13.39	22403	11417	242	31.2	31.81
PDS-30	16.62	14977	3316	3296	752.9	47.24	40097	20037	412	267.25	138.26
PC-BO-1	471.27	6868	1482	132	24.81	117.57	29664	24301	99	96.82	95.61
PC-BO-4	2573.21	41587	3078	46	229.34	264.96	104695	67495	41	423.06	243.57
MO-MI-2	1238.57	38968	2004	105	374.06	457.69	101523	65220	59	1737.77	283.87

Table 1. Performance comparison. Solver used is Cplex 12.4 on AMD Opteron(tm) Processor 246, 64 bit, 2.00 GHz, 2 Gb RAM, running Ubuntu Linux 2.6.32-41-server. Times are in seconds.

Slow convergence seems to be the main reason behind the weak performances of approaches a.2 and a.3 when solving the PDS and MNET instances. As sampled in Figure 1, our analysis reveal a clear relationship between relative performances and instability of the dual iterates. That is, the dual variables (used to generate columns) oscillate wildly even close to convergence for the PDS and MNET instances, but not for the TTP ones. This well-known instability of column generation approaches, hereby a "good" dual solution is abandoned to move to a much worse one, considerably slows down the convergence.

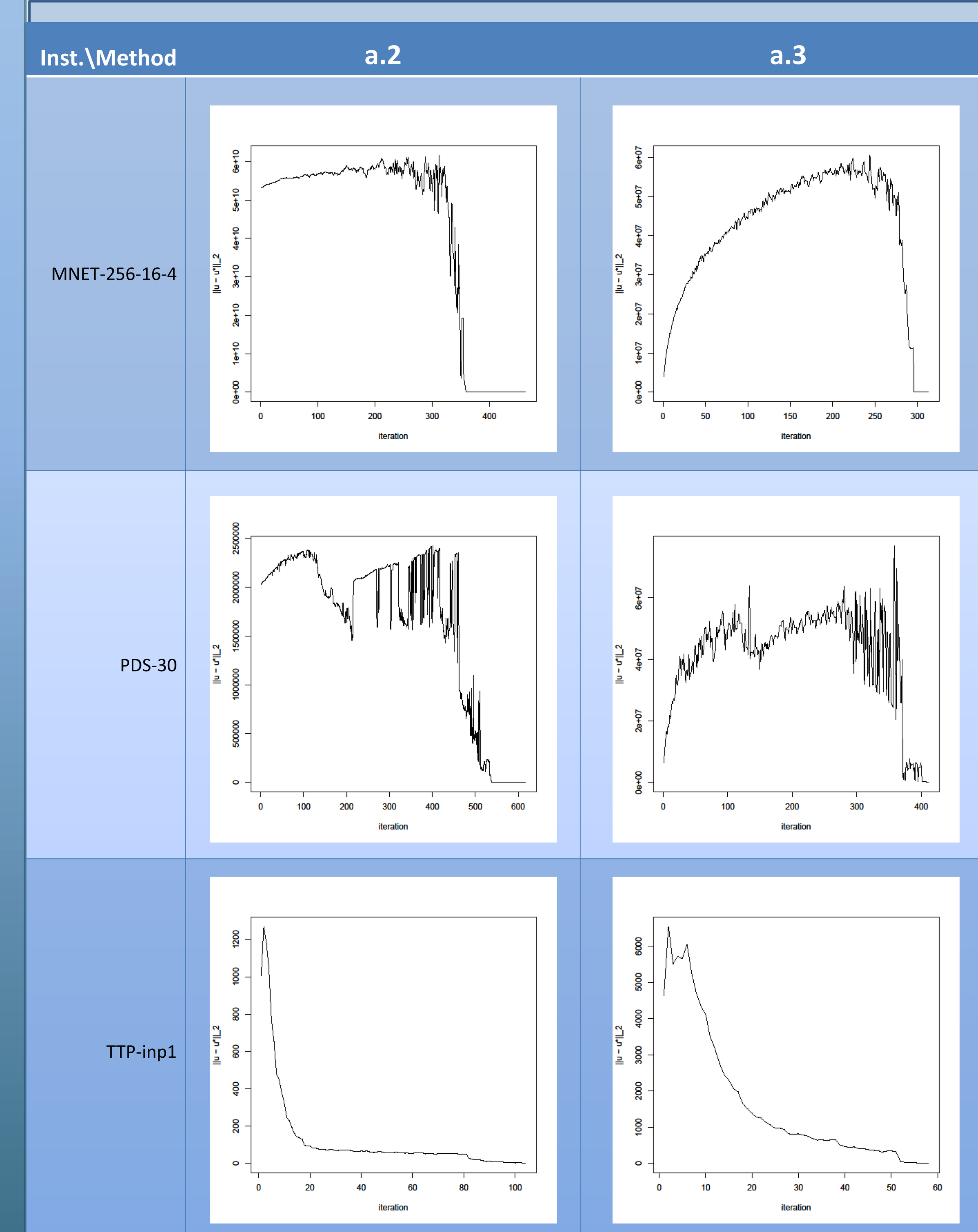
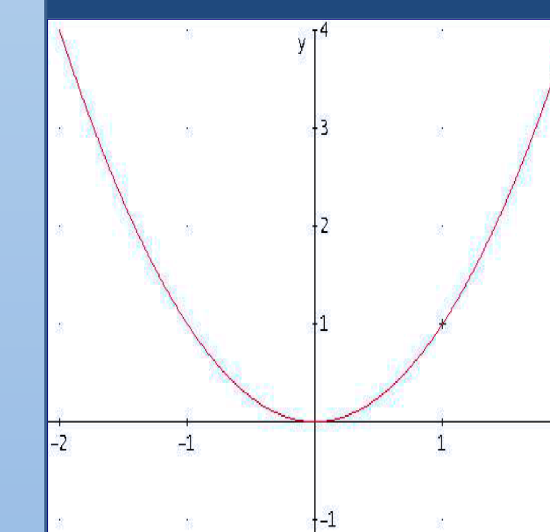


Figure 1. Distance $\| u - u^* \|_2$ between the dual variable u and the optimal point u^*.

FUTURE WORK

One device to cope with this instability problem consists in *stabilization* of the column generation, see, e.g. [1]. This has already been successfully applied to MCFP, albeit in an aggregated formulation [6,3] and with specialized approaches for solving the corresponding (quadratic) master problem [4], while non-quadratic stabilization is also possible [5] and has not been tested for MCFP. Furthermore, a.3 can be seen as a non-stabilized special case of the general class of approaches defined in [7], and therefore stabilization can be applied in that case, too. Hence, we intend to investigate if and how stabilization can make a.2 and a.3 competitive with a.1 on instances where they are currently not (like the PDS ones), and further improve their performances in these where they are competitive already (like the TTP ones).

Stabilization of the Column Generation



Stabilizing the column generation consists in penalizing the distance between the new dual solution and the best dual solution so far. In dual space, column generation can be seen as a *cutting plane* method. Thus, stabilization can be understood as a cutting plane acceleration scheme. Indeed the column generation stabilization is a special case of the *bundle method*.

REFERENCES

1. H. Ben Amor, J. Desrosiers, and A. Frangioni. On the Choice of Explicit Stabilizing Terms in Column Generation. Discrete Applied Mathematics, 157(6):1167-1184, 2009.
2. Valentina Cacchiani, Alberto Caprara, and Paolo Toth. A column generation approach to train timetabling on a corridor. 4OR: A Quarterly Journal of Operations Research, 6:125-142, 2008. 10.1007/s10288-007-0037-5.
3. P. Cappanera and A. Frangioni. Symmetric and Asymmetric Parallelization of a Cost-Decomposition Algorithm for Multi-Commodity Flow Problems. INFORMS Journal on Computing, 15(4):369-384, 2003.
4. A. Frangioni. Solving Semidenite Quadratic Problems Within Nonsmooth Optimization Algorithms. Computers & Operations Research, 21:1099-1118, 1996.
5. A. Frangioni. Generalized Bundle Methods. SIAM Journal on Optimization, 13(1):117-156, 2002.
6. A. Frangioni and G. Gallo. A Bundle Type Dual-Ascent Approach to Linear Multicommodity Min Cost Flow Problems. INFORMS Journal on Computing, 11(4):370-393, 1999.
7. A. Frangioni and B. Gendron. A Stabilized Structured Dantzig-Wolfe Decomposition Method. Mathematical Programming, to appear, 2012.
8. John W. Mamer and Richard D. McBride. A decomposition-based pricing procedure for large-scale linear programs: An application to the linear multicommodity Flow problem. Management Science, 46(5):pp. 693-709, 2000.

CONTACTS

Antonio Frangioni: Room 381 DO, Dipartimento di Informatica, Largo B. Pontecorvo, 3 - 56127 Pisa, Italy. Tel: +39 050 2212789.

E-mail: frangio@di.unipi.it.

Tiziano Parriani: Dipartimento di Elettronica Informatica e Sistemistica, Viale Risorgimento, 2 - 40136 Bologna, Italy.

E-mail: tiziano.parriani@unibo.it