

# A Novel Lagrangian-relaxation to the Minimum Cost Multicommodity Flow Problem and its Application to OSPF Traffic Engineering

Gábor Rétvári, József J. Bíró, Tibor Cinkler  
 High Speed Networks Laboratory, QoSIT Laboratory  
 Department of Telecommunications and Media Informatics  
 Budapest University of Technology and Economics  
 H-1117, Magyar Tudósok körútja 2., Budapest, HUNGARY  
 Email: {retvari,biro,cinkler}@tmit.bme.hu.

**Abstract**—The Minimum Cost Multicommodity Flow problem plays a central role in today's operations research theory with applications ranging from transportation and logistics to telecommunications network routing. In this paper, we introduce a novel Lagrangian-relaxation technique, which, given an initial feasible solution, can solve the minimum cost multicommodity flow problem as a sequence of single-commodity flow problems. Our methodology is best suited for OSPF traffic engineering, because it can rapidly improve a given path set towards approximate optimality while simultaneously provides the link weights, which implement the paths as shortest paths.

## I. INTRODUCTION

Let  $G(V, E)$  be a directed graph, formed by the set of nodes  $V$  ( $|V| = n$ ) and the set of edges  $E$  ( $|E| = m$ ). Let the capacity of an edge  $(i, j)$  be given by  $u_{ij} > 0$  and the cost by unit of traffic be given by  $c_{ij} > 0$ . Let the capacity and cost values be gathered into  $m$ -vectors, say,  $u$  and  $c$ , respectively. Let  $K$  denote the set of source-destination pairs  $(s_k, d_k) : k \in K$ , which are referred to as *sessions* for short. There is  $t_k$  amount of demand associated with each session  $k$ . In vector notation,  $t^k$  is an  $n$ -vector, such that  $(t^k)_i = t_k$  if  $i = d_k$ ,  $(t^k)_i = -t_k$  if  $i = s_k$  and all zero otherwise. Let the arc flow sent by session  $k$  to edge  $(i, j)$  be denoted by  $X_{ij}^k$  and let us gather the values of  $X_{ij}^k$  for session  $k$  into the  $m$ -vector  $x^k$ . Finally, let  $N$  be the node-arc incidence matrix associated with  $G$  [2]. Then, we consider the minimum cost multicommodity flow problem in the following classical form:

$$\min \sum_{k \in K} c x^k \quad (1)$$

$$N x^k = t^k \quad \forall k \in K \quad (2)$$

$$\sum_{k \in K} x^k \leq u \quad (3)$$

$$x^k \geq 0 \quad \forall k \in K \quad (4)$$

In this formulation, the objective (1) is to minimize the aggregate cost of arc flows  $x^k$  summed over all sessions  $k$  and all edges  $(i, j)$ , while satisfying the so called *flow conservation* constraints (2) for each commodity at each node. The independent single-commodity flow problems are coupled together by the *bundle constraints* (3), which require that the capacity of a link must not be violated by the sum of the flows

of all commodities on that link. Finally, the *non-negativity* constraints in (4) specify that the flows are non-negative.

The linear programming formulation of the minimum cost multicommodity flow problem contains one variable for each commodity for each edge (a total number of  $km$ ) and a flow conservation constraint for each node and each commodity ( $kn$ ) plus a bundle constraint for each link summing up to a total number of  $kn + m$  constraints. Hence, the size of the linear problem grows dramatically with the increase of the network and user population, and even for a medium sized problem the ubiquitous simplex algorithm merely crawls towards the optimum. As of our favorite open source linear programming toolkit, the GNU Linear Programming Kit (GLPK, [1]), for a network of 60 nodes, 180 edges and 35 sessions the solution process takes almost five minutes on a 800 MHz Intel PIII processor and consumes more than 12 MB of RAM. Therefore, various decomposition techniques, approximations and relaxations are proposed in the literature to facilitate fast solution of large-scale multicommodity flow problems (for a comprehensive evaluation of these methods, consult [2], [3] and [4] and references therein).

Nevertheless, neither column-generation nor basis partitioning methods promise to speed up the solution process by more than one order of magnitude. This is still completely inconvenient for traffic engineering, which requires rapid algorithms to ensure quick adaptation to topology changes or management controls. Therefore, in this paper we propose a new Lagrangian-relaxation method to iteratively solve the minimum cost multicommodity flow problem starting from an initial feasible solution, permitting us to reduce memory requirements and trading-off running time for the precision of the solution. A large number of Lagrangian-relaxation techniques for solving the minimum cost multicommodity flow problem is known in the literature [2], however, our formulation is special, as it has some very interesting consequences and is specifically tailored to OSPF traffic engineering.

In OSPF traffic engineering, we are given a set of sessions and a set of paths for each session. The task is to either decide that the paths are not representable as shortest paths, or

otherwise compute appropriate link weights as to implement the paths as shortest paths between the endpoints of the sessions. This problem is of crucial importance in the vast majority of today's IP networks, which still mostly rely on shortest path routing protocols, such as OSPF [5]. OSPF traffic engineering promises to increase the network revenue and in some settings it may provide performance close to optimal exploiting the previously hidden capabilities of legacy routing hardware and software [6], [7], [8]. A groundbreaking work of Wang *et al.* [9] concludes that a set of paths is either reproducible as shortest paths, or it can be improved upon and transformed into a set of paths of the same capacity yet consuming less bandwidth. The improved path set is in turn shortest path reproducible. This can be done by solving a minimum cost multicommodity flow problem starting from an initial feasible solution, which is exactly what the proposed Lagrangian-relaxation technique is best suited for.

The rest of this paper is structured as follows. In Section II we discuss the proposed Lagrangian-relaxation. Section III reveals, how to use our method for the purpose of OSPF traffic engineering and Section IV briefly presents some related simulation results. Finally, in Section V we draw the conclusions of our work.

## II. THE IMPROVEMENT PROBLEM

Assume that we already know some feasible solution to the multicommodity flow problem. Let the feasible solution be given by the arc flow vector  $y^k = [Y_{ij}^k] : (i, j) \in E$  for each session  $k \in K$ . Then,  $y^k$  solves the constraint system of (1)-(4), i.e.,  $\{\forall k \in K : Ny^k = t^k, \sum_{k \in K} y^k \leq u, y^k \geq 0\}$ . Furthermore, consider the following linear program, the so called Improvement Problem (P-IMP):

$$\min \sum_{k \in K} cx^k \quad (5)$$

$$Nx^k = 0 \quad \forall k \in K \quad (6)$$

$$\sum_{k \in K} x^k \leq u - \sum_{k \in K} y^k \quad (7)$$

$$x^k \geq -y^k \quad \forall k \in K \quad (8)$$

The improvement problem possesses special structure. For example, constraints (6) define flow circulations (if  $x^k \neq 0$ ) of capacity circumscribed by constraints (7) and (8). Next, we show that, given a feasible solution  $y^k$ , solving P-IMP equals to solving the minimum cost multicommodity flow problem. Furthermore, due to its special structure, P-IMP lends itself to price-directive decomposition, which has some notable consequences.

*Theorem 1:* Let an instance of the minimum cost multicommodity flow problem be  $I$  and let a feasible solution to  $I$  be given by  $y^k$ . Furthermore, let an optimal solution of the corresponding improvement problem be  $x^k$ . Then,  $z^k = y^k + x^k$  is an optimal solution of  $I$ .

*Proof:* In order to prove the theorem it is enough to show that arc flow vectors  $z^k$  satisfy flow conservation constraints (2), bundle constraints (3) and non-negativity constraints (4). Then, any optimal solution to P-IMP, by definition, minimizes  $I$  as well, because the objectives only differ in a constant term

$\sum_{k \in K} cy^k$ . The flow conservation constraints (2) obviously hold, since

$$Nz^k = N(y^k + x^k) = Ny^k + Nx^k = t^k + 0,$$

and because  $y^k$  is feasible,  $z^k$  is also feasible. The bundle constraints (3) also hold, because

$$\sum_{k \in K} z^k = \sum_{k \in K} (x^k + y^k) \leq u$$

according to (7). Finally,  $z^k \geq 0$ , as  $y^k + x^k \geq 0$  according to (8), which completes the proof. ■

By introducing the improvement problem, the minimum cost multicommodity flow problem is reformulated as a linear optimization problem. In this problem, the task is to find some flow circulations, which, together with the initial feasible solution  $y^k$  give a minimum cost multicommodity flow instance. Recall from network flow theory that any feasible single-commodity flow instance can be transformed into any other feasible flow instance along at most  $m$  directed circulations [2]. This leads to a very interesting corollary of Theorem 1:

*Corollary 1:* Any feasible multicommodity flow instance can be transformed into any other feasible multicommodity flow instance along at most  $km$  single-commodity flow circulations.

Note that in its linear programming formulation even P-IMP poses the same difficulties as (1)-(4). A well known technique to solve large-scale optimization problems is the Lagrangian-relaxation method [3], [4]. Lagrangian-relaxation permits us to iteratively solve complex problems by consecutively optimizing the embedded network flow problems as simple uncoupled problems. Generally speaking, consider the problem instance:

$$I := \min\{cx : Ax \leq b, x \in \mathcal{N}\},$$

where  $\mathcal{N}$  is a specially constrained set, e.g., it may be a convex space spanned by feasible solutions of some network flow problem. The task is then to minimize an objective function  $cx$  over  $\mathcal{N}$ , such that the resultant  $x$  vector satisfies the side constraints  $Ax \leq b$ . Such optimization problems tend to be rather hard to solve, and may very well be NP hard. The *Lagrangian dual problem* is defined as:

$$L(w^*) = \max_{w \geq 0} L(w),$$

where

$$L(w) = \min_{x \in \mathcal{N}} \{cx + w(Ax - b)\}$$

is called the *Lagrangian-relaxation* of  $I$ . In this case, the Lagrangian-relaxation solves  $I$  if for some choice of the Lagrange multipliers  $w^*$  the solution of the Lagrangian-relaxation  $x^*$  is feasible in  $I$  and satisfies the complementary slackness conditions  $w^*(Ax^* - b) = 0$ . Note that  $x \in \mathcal{N}$  vectors, by assumption, are easy to generate. Hence, various nonsmooth optimization techniques are known to solve the Lagrangian dual problem iteratively, such as the subgradient method, bundle methods and cutting-plane methods [4].

A straightforward Lagrangian-relaxation of the minimum cost multicommodity flow problem (1)-(4) arises if one incorporates the bundle constraints (3) into the objective function [2]. This technique is often referred to as price-directive decomposition. A very interesting property of the improvement problem is that it also lends itself to price-directive decomposition. Given an initial feasible solution  $y^k$ , one obtains the following new Lagrangian-relaxation to the minimum cost multicommodity flow problem:

$$L(w) = \min \sum_{k \in K} cx^k + w(\sum_{k \in K} x^k + \sum_{k \in K} y^k - u) \quad (9)$$

$$Nx^k = 0 \quad \forall k \in K \quad (10)$$

$$x^k \geq -y^k \quad \forall k \in K \quad (11)$$

Then, the Lagrangian dual problem can be written as:

$$L(w^*) = \max_{w \geq 0} \sum_{k \in K} L^k(w) + w(\sum_{k \in K} y^k - u),$$

and the Lagrangian-relaxation decomposes into independent Lagrangian subproblems P-SUBIMP for each commodity  $k \in K$ :

$$L^k(w) = \min(c + w)x^k \quad (12)$$

$$Nx^k = 0 \quad (13)$$

$$x^k \geq -y^k \quad (14)$$

Now, we make the following observations. Since constraint (13) define flow circulations, a cycle cancellation algorithm is probably a good choice to solve P-SUBIMP. We only need to assure that the cycles carry the most negative flow possible while satisfying (14) as well. In fact, an arc flow  $X_{ij}^k$  is only circumscribed by the initial flow  $Y_{ij}^k$  of the same commodity. If, for some edge  $(i, j)$ ,  $Y_{ij}^k = 0$ , then  $X_{ij}^k \geq 0$  holds and we can not improve the objective function on that link. If  $Y_{ij}^k \geq 0$ , then  $X_{ij}^k$  is required to be the most negative possible (i.e.,  $-Y_{ij}^k$ ). A well known way to handle negative flows (and many other issues) is the notion of residual graphs: In a residual graph, negative flow on edge  $(i, j)$  is represented by a positive flow on the associated reverse edge  $(j, i)$ . Hence, constraint  $X_{ij}^k \geq -Y_{ij}^k$  can be written as  $X_{ji}^k \leq Y_{ij}^k$ , which just corresponds to the residual capacity on  $(j, i)$  after sending  $Y_{ij}^k$  amount of flow on  $(i, j)$ . The negativity of the reverse flow is captured by associating a negative cost  $c_{ji} + w_{ji} = -(c_{ij} + w_{ij})$  with the reverse edges (for more coverage on residual graphs, the reader is referred to the precious discussion in [2]). Hence, searching for the most negative-valued flow circulations in  $G$  over positive edge costs  $c_{ij} + w_{ij}$  equals to searching for the *most negative-cost positive-valued flow circulation* in the residual graph  $G(y^k)$  induced by the initial flow  $y^k$ . This task can be accomplished by the minimum mean cycle cancellation algorithm in  $O(n^2 m^2 \log(nC))$  time, where  $C$  is the largest cost value [10], [11]. We summarize the above discussion as follows:

**Observation 1:** Consider the residual graph  $G(y^k)$  obtained by instantiating  $y^k$  in the *uncapacitated* network  $G^\infty$ . Then, solving the improvement problem P-SUBIMP in  $G$  equals to

solving the minimum cost single-commodity flow circulation problem over the modified cost set  $c_{ij} + w_{ij}$  in  $G(y^k)$ .

At this point, we have basically discovered a universal methodology to improve an initial feasible solution of an optional multicommodity flow problem instance till it becomes optimal. From Corollary 1 we know that any initial feasible solution  $y^k$  can be transformed into the optimal solution along at most  $km$  flow circulations. Such flow circulations can be found by executing an independent cycle cancellation process for each commodity over a set of optimal Lagrange multipliers.

It is a useful property of Lagrangian-relaxation that the optimal Lagrange multipliers  $w^*$  are dual-optimal. This means that for an edge  $(i, j)$ , an optimal Lagrange multiplier  $w_{ij}^*$  equals to the optimal dual variable associated with the bundle constraint of the improvement problem for that link. However, as the next theorem shows, we can say even more about the optimal Lagrange multipliers  $w^*$ :

**Theorem 2:** Let  $I$  be an instance of the minimum cost multicommodity flow problem, and  $w^*$  be the set of Lagrange multipliers, which are optimal to the Lagrangian-relaxation formulation (9)-(11) of the corresponding improvement problem  $I'$ . Then,  $w^*$  are dual-optimal to both the improvement problem  $I'$  and the multicommodity flow problem  $I$ .

**Proof:** Consider an optimal primal solution  $z^k$  of  $I$  and a set of Lagrange multipliers  $w^*$  and flows  $x^k$ , which are optimal to  $I'$ . Then, we only need to show that  $z^k$  and  $w^*$  together satisfy the complementary slackness criteria:  $w^*(\sum_{k \in K} z^k - u) = 0$ . Recall that after solving the improvement problem  $I'$  to optimality, one obtains the solution of  $I$  in the form:  $z^k = y^k + x^k$ . However, since  $x^k$  is optimal with respect to the initial feasible solution  $y^k$ , so the complementary slackness criteria for  $I'$  can be written as:  $w^*(\sum_{k \in K} x^k + (\sum_{k \in K} y^k - u)) = w^*(\sum_{k \in K} (x^k + y^k) - u) = 0$ . This proves the theorem. ■

### III. AN APPLICATION TO OSPF TRAFFIC ENGINEERING

In this section, we introduce some more mathematical notation to formulate the principal theorem of shortest path representability, i.e., the property of a set of paths that it can be represented as shortest paths by some positive link weights. Then, we reveal how to use the Lagrangian-relaxation technique developed in the previous section to not only make a decision on the shortest path representability of a particular path set, but to also calculate the optimal link weights in one turn.

Let  $\mathcal{P}^{s_k \rightarrow d_k}$  be the set of all paths that connect a particular source-destination pair  $(s_k, d_k)$  in  $G$ . In OSPF traffic engineering our task then is to explicitly represent a given subset  $\mathcal{P}^k \subseteq \mathcal{P}^{s_k \rightarrow d_k}$  as shortest paths for each session  $k \in K$ . Let  $\mathcal{P} = \bigcup_{k \in K} \mathcal{P}^k$ ,  $n^k = [n_{ij}^k] : (i, j) \in E$  be the number of paths of session  $k$  traversing link  $(i, j)$  and  $n_{ij} = \sum_{k \in K} n_{ij}^k$ . The network induced by  $\mathcal{P}$  is the so called path-graph  $G_{\mathcal{P}}$ .  $G_{\mathcal{P}}$  includes all edges of all paths of  $\mathcal{P}$  and the capacity of the edges equals to the number of paths in  $\mathcal{P}$  using that link, i.e.,  $u_{ij} = n_{ij}$ . We associate  $p_k = |\mathcal{P}^k|$  amount of demand with each session in the path-graph. We assume that  $G_{\mathcal{P}}$

is strongly connected with respect to the session endpoints  $(s_k, d_k) : k \in K$ . Then, the principal theorem of shortest path representability can be stated as follows:

*Proposition 1:* Let  $I$  be an instance of the minimum cost multicommodity flow problem, which is characterized by the path graph  $G_{\mathcal{P}}$  induced by a set of paths  $\mathcal{P}$ . Furthermore, let the link costs be  $c_{ij} = 1$  and the demand set be  $t_k = p_k$ . Let a solution to  $I$  be  $y^k = n^k$ . Note that  $y^k$  is obviously feasible but not necessarily optimal to  $I$ . Then, the path set  $\mathcal{P}$  is shortest path representable if and only if  $y^k$  is optimal to  $I$  [9].

The above Proposition has far-reaching consequences, which helped to disprove the common belief of researchers that shortest path routing is, by nature, useless to traffic engineering. Namely, it is relatively easy to show that a set of paths is either representable as shortest paths, or it is loopy, and hence, is of negligible interest to traffic engineering. In the latter case, the optimal solution to  $I$  defines a path set, which is of the same capacity as  $\mathcal{P}$  but uses strictly less bandwidth. Shortest path representability can be confirmed by solving  $I$ , or solving the following dual of  $I$ :

$$\max \sum_{k \in K} p_k \pi_{d_k}^k - \sum_{(i,j) \in E} n_{ij} w_{ij} \quad (15)$$

$$\pi_j^k - \pi_i^k \leq 1 + w_{ij} \quad \forall (i,j) \in E, \forall k \in K \quad (16)$$

$$w_{ij} \geq 0 \quad \forall (i,j) \in E \quad (17)$$

An interesting property of this dual formulation is that it also supplies link weights in the form  $1 + w_{ij}$  together with node potentials  $\pi_n^k$ , which represent  $\mathcal{P}$  as shortest paths. This is the corollary of the complementary slackness property:  $n_{ij}^k > 0 \Rightarrow \pi_j^k - \pi_i^k = 1 + w_{ij}$  [9]. Note that both primal and the dual formulation involves an enormous number of variables and constraints even for medium-sized problems, so solving it by the simplex method puts merely unbearable CPU and memory burden on network devices. Therefore, we propose to use the Lagrangian-relaxation technique described in Section II to find an optimal solution iteratively in the following way:

First, as  $n^k$  is feasible to  $I$ , one can use it as an initial feasible solution. This can be improved upon by solving P-IMP, or equivalently, consecutively solving the Lagrangian-subproblem (12)-(14) with  $y^k = n^k$  until optimality. In light of Observation 1, this can be done easily by the minimum mean cycle-cancellation algorithm.

A convenient and easy-to-use (but not necessarily the most efficient [4]) way to solve the corresponding Lagrangian dual problem  $L(w^*) = \max_{w \geq 0} \sum_{k \in K} L^k(w) + w(\sum_{k \in K} n^k - u)$  is the subgradient method. In the  $n$ th iteration step of the subgradient method, we execute the minimum mean cycle cancellation algorithm for each session  $k$  in the residual graph  $G(y^k)$  characterized by the link weights  $1 + w_{ij}^n$ . If the resultant  $z^k = n^k + x^k$  arc flow set is primal feasible and satisfies the complementary slackness criteria, we stop the iteration and conclude that the current solution  $z^k$  is optimal. If, at any iteration, the complementary slackness criteria are violated but  $z^k$  is primal feasible and improves the objective function, we

first conclude that, according to Proposition 1, the given set of paths is not shortest path representable. Optionally, we can restart the subgradient method from this feasible solution to obtain an improved path set. This has the potential to reduce the amount of work the cycle-cancellation algorithm has to do at each iteration step since the new initial solution is “closer” to the optimum. Otherwise, we modify the Lagrange multipliers  $w_{ij}^{n+1} = (w_{ij}^n + \frac{1}{n} \sum_{k \in K} X_{ij}^k)^+$  and execute step  $n + 1$ .

Furthermore, we observe that Theorem 2 has a very interesting consequence related to shortest path representability, which we formulate as follows:

*Corollary 2:* The optimal Lagrange multipliers  $w_{ij}^*$  are dual optimal to  $I$ . Hence,  $w_{ij}^*$  solve the dual of  $I$ , and therefore  $1 + w_{ij}^*$  define positive weights, which represent the given path set as shortest paths.

The proposed method has a number of advantages in comparison to standard linear programming solution methods. Principally, Lagrangian-relaxation reduces the amount of necessary memory and CPU resources, because the uncoupled subproblems are much easier to solve than the full-fledged problem. What sets apart the proposed method from other Lagrangian-relaxations aimed at the same purpose is that, depending on the quality of the initial feasible solution, it is believed to be more efficient in the average case. This is because our method tries to improve a potentially close-to-optimal solution rather than starting from scratch in each step (which would be the case for instance with the shortest path Lagrangian-relaxation). If a better feasible solution is found, our method uses that solution as the new starting point, which definitely speeds up the process. In addition, Corollary 2 assures that running the subgradient method to optimum yields some link weights, which represent the optimal path set as shortest paths. However, this may be a tedious task and may very well last more than running a standard linear program solver. Nevertheless, since the subproblems can be solved in strictly polynomial time, the total time taken by the optimization can be limited by stopping the process after a pre-defined number of iterations. As the following brief discussion of some related simulation studies suggests, the resultant *approximate OSPF Traffic Engineering* scheme may also very well be of large practical interest.

#### IV. SIMULATION RESULTS

Our simulations were aimed at comparing the performance and the execution time of the simplex method and the proposed Lagrangian-relaxation technique. An intriguing question one faces when attempting to evaluate the performance of a particular OSPF traffic engineering method is how to obtain a high quality traffic-engineered path set. In light of the fact that this problem is NP hard [12], we decided to use the paths yielded by the optimal solution of the maximum throughput problem. Our experiments with other algorithms suggest that it is the maximum throughput problem, which gives a reasonably good path set in the majority of the random topologies.

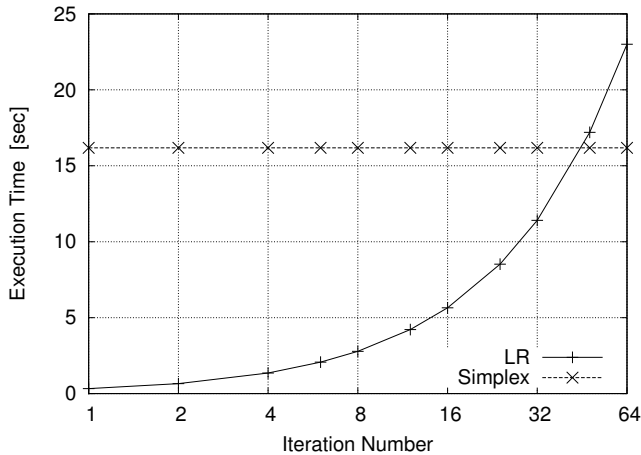


Fig. 1. Average execution time of the proposed Lagrangian-relaxation (LR) as the function of the iteration count compared to the execution time of the simplex method

The results presented below are averaged over 50 simulation rounds. In each round, we generated a random graph of 40 nodes, 120 edges. We also instantiated 18 sessions in the network at random locations. We calculated an initial path set in the following way: First, we solved the maximum throughput problem to obtain a feasible demand set. This demand set defines the maximum amount of traffic the network can serve over the actual topology and the placement of session endpoints. Then, we set all link costs  $c_{ij} = 1$  and solved the *maximum* cost multicommodity flow problem over the demand set. This produces a path set (and a path-graph), which is the furthest possible from the optimal one (which would be the result of solving the *minimum* cost problem), and used that as an initial feasible solution to start our Lagrangian-relaxation from. We expect the method to develop the path set towards optimality and, at each iteration, produce an improved set of link weights. We also used the simplex method to solve the dual minimum cost multicommodity flow problem to obtain precise traffic engineered link weights.

Figure 1 depicts the time taken by executing a pre-defined number of iterations of the subgradient method. Note that our implementation is only a “proof-of-concept” code written mostly in Perl and by no means meant to be overly efficient. This makes the comparison with the performance-optimized simplex implementation substantially pessimistic, however, even our suboptimal code is significantly faster at low iteration numbers. A more equitable comparison would be to consider the iteration count: The simplex method generally performs 1000-1500 iterations, while our method usually produces reasonable link weights in no more than a few dozen iterations (see below). Additionally, the amount of memory required by our method never exceeded the magnitude of some ten kilobytes, while the simplex method ordinarily consumes at the scale of megabytes (the use of column-generation would be especially helpful here).

We also fed the link weights produced after a particular

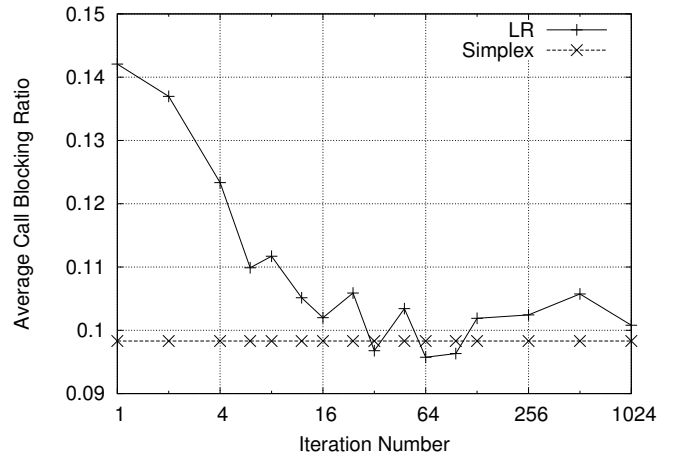


Fig. 2. Average call blocking ratio (CBR) when using the link weights produced by the proposed Lagrangian-relaxation (LR) as the function of the executed iterations compared to the CBR yielded by simplex link weights

number of iterations into a call-level OSPF simulator to measure the average call blocking ratio (Figure 2). Our results suggest that performance close to optimal (which is yielded by the exact link weights computed by the simplex method) can be achieved by as few as some 10-20 iterations and it is not worth running more than a hundred rounds. The mere fact that we were able to optimize a pretty bad set of paths in barely a few dozen iterations underlines the potential practical usefulness of the proposed Lagrangian-relaxation technique.

## V. CONCLUSIONS

In this paper, we introduced a novel Lagrangian-relaxation methodology, which, given an initial feasible solution, can iteratively solve the minimum cost multicommodity flow problem. The proposed method may be of general interest, not only because it solves a key problem in operations research but because it is very descriptive and has some fundamental corollaries related to the transformability of multicommodity flow instances. We showed that our Lagrangian-relaxation method is particularly useful in the field of OSPF traffic engineering. On the one hand, given a set of paths, our method can rapidly decide, whether or not the path set is representable as shortest paths. Furthermore, it quickly provides approximate link weights to implement the shortest path representation without the need to solve large-scale resource-hungry linear programs. The full-fledged problem is decomposed into uncoupled minimum cost flow circulation subproblems, for which implementations and practical experience are readily available. On the other hand, as the computed link weights are not necessarily integer-valued, some post-processing work may be desirable before they can be conveniently used in real-life routing protocols. An obvious way to overcome this problem would be to use constant step size in the subgradient optimization, however, this raises convergence issues. Our further efforts will be focused on this problem.

## REFERENCES

- [1] The GLPK project: <http://www.gnu.org/software/glpk/glpk.html>.
- [2] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Englewood Cliffs, NJ, 1993.
- [3] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali, *Linear Programming and Network Flows*. John Wiley & Sons, January 1990.
- [4] C. Lemaréchal, "Lagrangian relaxation," in *Computational Combinatorial Optimization* (M. Jünger and D. Naddef, eds.), pp. 115–160, Springer Verlag, Heidelberg, 2001.
- [5] J. Moy, "OSPF Version 2." RFC 2328, April 1998.
- [6] B. Fortz, J. Rexford, and M. Thorup, "Traffic engineering with traditional IP routing protocols," *IEEE Communications Magazine*, vol. 40, pp. 118–124, Oct 2002.
- [7] B. Fortz and M. Thorup, "Optimizing OSPF/IS-IS weights in a changing world," *IEEE Journal of Selected Areas in Communications*, vol. 20, pp. 756–767, May 2002.
- [8] A. Sridharan, C. Diot, and R. Guérin, "Achieving near-optimal traffic engineering solutions for current OSPF/IS-IS networks," in *Proceedings of INFOCOM 2003*, March 2003.
- [9] Z. Wang, Y. Wang, and L. Zhang, "Internet traffic engineering without full-mesh overlaying," in *Proceedings of INFOCOM 2001*, April 2001.
- [10] R. Karp, "A characterization of the minimum cycle mean in a digraph," *Discrete Mathematics*, vol. 23, pp. 309–311, 1978.
- [11] A. Goldberg and R. Tarjan, "Finding minimum-cost circulations by canceling negative cycles," *J. ACM*, vol. 36, no. 4, pp. 873–886, 1989.
- [12] G. Rétvári and R. Szabó, "On the representability of arbitrary path sets as shortest paths: Theory, algorithms and complexity." to appear in the Proceedings of IFIP Networking 2004.