A Fully Distributed Lagrangean Metaheuristic for a P2P Overlay Network Design Problem[*]

Marco A. Boschetti[†] Márk Jelasity[*] Vittorio Maniezzo[*]

[*]Department of Computer Science, University of Bologna
Mura Anteo Zamboni 7, I-40126 Bologna, Italy
{maniezzo,jelasity}@cs.unibo.it

[†]Department of Mathematics, University of Bologna
Piazza di Porta San Donato 3, I-40126 Bologna, Italy
boschett@csr.unibo.it

1 Introduction

In peer-to-peer (P2P) networks it is a central problem to maintain a so called *overlay network* with certain desired properties. An overlay network is defined by logical connections (i.e., the "who knows whom" relation) between peers over an underlying physical network. If node i is connected to node j in an overlay network, it means that node i knows the address of j and so it can send messages to j. An overlay network must fulfill certain requirements to allow for optimal cost and efficiency of the application of the overlay. Besides, a typical P2P network is large, heterogeneous and very dynamic, which makes the overlay network construction problem even harder. In this work we address the Membership Overlay Problem (MOP) [8], a special case of the general overlay network construction problem. In this problem, we are interested in constructing an overlay network which is *unstructured*, that is, used to define the membership of a dynamic set of peers. Unstructured overlay networks have many important applications such as information dissemination and data aggregation (datamining) [3, 7]. In this case, each node sends gossip messages periodically to its neighbors. It is important that load is distributed in a fair manner so that the throughput of the network is maximized without any nodes being overloaded.

The MOP can be formulated as follows. A graph $G = (V, E)$ of n vertices is given, where the nodes correspond to peers that want to communicate with each other. The edges correspond to possible communications, i.e., if there is an edge (i, j) then i can possibly send a message to j using the underlying routing infrastructure. Each node can dynamically enter and exit the network, and when it is connected it can make use of a limited bandwidth. Therefore, each

[*]Márk Jelasity is also with RGAI, MTA SZTE, Szeged, Hungary. This work was partially supported by the Future & Emerging Technologies unit of the European Commission through Project BISON (IST-2001-38923).

node has two associated weights, p_i and w_i, $i = 0, \ldots, n$, corresponding to its uptime (measured as the percentage of time that the peer is available and responding to traffic, normalized to 1) and to the available bandwidth of its connection to the Internet, respectively.

The MOP asks to find a subgraph $G' = (V, E')$ of G. The edges in the graph G' define the fact that two nodes actually decide to allocate some bandwidth to communicate with each other, i.e., when two nodes i and j establish a connection, each one must allocate part of its bandwidth. If b_i and b_j are the bandwidths which could be allocated by i and j, then the bandwidth of the connection can be at most $b_{ij} = min\{b_i, b_j\}$. The two values b_i and b_j could be equal to w_i and w_j or could be less than that, due to other connections already maintained by the peers. Moreover, there is a lower bound l_{ij} on the bandwidth of acceptable connections and a limit on the maximal value u_{ij} that b_{ij} can take. The graph G' has to be such that the expected network throughput is maximized, the diameter of G' is kept logarithmic and the total bandwidth used by each node i is less than or equal to b_i.

As mentioned, the algorithm for solving this problem should be local: no global knowledge of the network is provided, each node i can exchange information only with the nodes in N'_i, that is, with its neighbors in G'. Preliminary work on this problem was reported in [8].

1.1 The static problem

First a mathematical formulation (P) of the static version of the problem will be presented, which will be later adapted to the dynamic case. A comprehensive mathematical analysis of the MOP can be found in [2]. Two sets of decision variables are used: $\{x_{ij}\}$ and $\{\xi_{ij}\}$, $(i, j) \in E$. The decision variables x_{ij} specify the bandwidth allocated to the connection between peers i and j. Therefore they are continuous variables $0 \le x_{ij} \le u_{ij}$, which will be further constrained when they are not 0 to be at least l_{ij}. Decision variables ξ_{ij} are binary variables which are 1 if arc (i, j) is used for a connection, 0 otherwise. The formulation, denoted by P, is the following:

$$z_P = max \sum_{(i,j) \in E} p_{ij} x_{ij} \tag{1}$$

$$s.t. \sum_{j \in N_i} x_{ij} \le b_i, \qquad i \in V \tag{2}$$

$$\sum_{i \in S_h} \sum_{j \in V \setminus S_h} \xi_{ij} \ge 1, \qquad \forall S_h \subset 2^V \tag{3}$$

$$l_{ij}\xi_{ij} \le x_{ij} \le u_{ij}\xi_{ij}, \qquad (i, j) \in E \tag{4}$$

$$\xi_{ij} \in \{0, 1\}, \qquad (i, j) \in E \tag{5}$$

where $p_{ij} = p_i \times p_j$, for each edge $(i, j) \in E$, and N_i represents the neighborhood of i in G (i.e., $N_i = V \setminus \{i\}$ if graph G is complete). Constraints (3) enforce connectivity as they require any subset of nodes to be connected to the subset of remaining nodes. Problem P is NP-hard [8].

The LP-relaxation of problem P is obtained by replacing constraints (5) with constraints in the form $0 \le \xi_{ij} \le 1$, for each $(i, j) \in E$. The resulting problem LP can be solved by an LP-solver provided that constraints (3), which are in exponential number, are added in a cutting plane fashion. However, in a practical setting and especially in dynamic scenarios, when nodes continuously leave and join the network, there will be other mechanisms to ensure connectivity of the overlay network, so it is not critical that our optimization framework enforces connectivity. Since in this abstract we focus on such scenarios, we omit constraints (3) from now on. We denote the resulting problem with P', which is still NP-hard as it subsumes the multiple knapsack problem.

Formulation P' can be effectively solved by a Lagrangian relaxation, associating non negative penalty λ_i to each constraint (2) and obtaining the following problem LR:

$$z_{LR}(\lambda) = max \sum_{(i,j)\in E} p'_{ij}x_{ij} + \sum_{i\in V} b_i\lambda_i \tag{6}$$

$$s.t.\ 0 \leq x_{ij} \leq u_{ij}, \qquad (i,j) \in E \tag{7}$$

where $p'_{ij} = p_{ij} - \lambda_i - \lambda_j$ and variables $\{\xi_{ij}\}$ are not required since, given λ, the optimal value $z_{LR'}(\lambda)$ can be computed according the following observations:
1) if $p'_{ij} \geq 0$ we use as much bandwidth as possible, i.e. $\xi_{ij} = 1$ and $x_{ij} = u_{ij}$;
2) if $p'_{ij} < 0$ we don't use the connection, i.e. $\xi_{ij} = 0$ and $x_{ij} = 0$.

In order to find the value of λ that minimizes the upper bound $z_{LR}(\lambda)$ we must solve the Lagrangian Dual $min\,[z_{LR}(\lambda) : \lambda \geq 0]$. The optimal solution of the Lagrangian Dual is equivalent to the optimal solution of the LP relaxation of problem P'.

2 A Lagrangean metaheuristic for the MOP

MOP was solved by means of a Lagrangean metaheuristic approach. Before detailing our approach, a general introduction is required.

In order to determine which algorithm is a metaheuristic and which is not, definitions are obviously needed. The naming was introduced by F.Glover, who denoted tabu search "as a 'meta-heuristic' superimposed on another heuristic" [5]. This definition was made more normative by stating that a metaheuristic "refers to a master strategy that guides and modifies other heuristics to produce solutions beyond those that are normally generated in a quest for local optimality" [6]. A discussion on this topic by S.Voss concludes that "A meta-heuristic is an iterative master process that guides and modifies the operations of subordinate heuristics to efficiently produce high-quality solutions" [9]. Maybe, one point which could still be underlined is included in this further definition of metaheuristic: "1) A high-level algorithmic framework or approach that can be specialized to solve optimization problems. or 2) A high-level strategy that guides other heuristics in a search for feasible solutions" [1], where the need of problem specialization is made explicit.

In the following we will outline a solution methodology based on the subgradient optimization [4] of the Lagrangean dual which, in its general structure, guides search for determining good quality solution of any combinatorial optimization problem and which, in order to be applied to a specific problem, must be specialized by defining a problem-specific repair procedure that permits obtaining (good quality) feasible solutions. Subgradient optimization is the master process while the Lagrangean heuristic is the subordinate heuristic. Therefore, we will outline a metaheuristic. To the best of our knowledge there were no previous efforts for framing subgradient optimization in a metaheuristic setting. We call this approach *Lagrangean metaheuristic.*

The basic idea is to use subgradient optimization for guiding search, and to repair each subgradient solution to make it feasible. Before introducing the details, we first remind that Lagrangean decomposition is a general approach for bounding the optimal cost of any combinatorial optimization problem. The central idea is that, whenever one has to solve a problem defined as $z^* = min\ cx$ subject to constraints $Ax = b$, $Cx = d$ and $x \in X$, where the first set of constraints is difficult to deal with, one can associate a *Lagrangean multiplier* (or *penalty*) to each constraint of the first set and solve the resulting Lagrangean subproblem $min\ \{cx + \mu(Ax - b) : Cx = d, x \in X\}$. Since any penalty value set yields a different problem instance, whose optimal cost is guaranteed to be a lower bound to z^*, we are actually facing a function $L(\mu)$ for which we want a maximum. Problem $L^* = max_\mu L(\mu)$ is referred to as the *Lagrangean Dual* of the initial problem. If $L(\mu^k)$ happens to be feasible for the original problem, it is also optimal. Otherwise, a *repair procedure* is needed to get an actual solution.

The complete pseudocode can be found in [2]. This approach has a number of advantages over most state-of-the-art metaheuristics. The main advantages are:
1) it is mathematically well-founded and it can rely over dozens of years of usage;
2) it includes optimality conditions to determine whether an optimum has been found;
3) it evolves both an upper and a lower bound to the problem to solve, thus at any time it can produce an estimation of the quality of the currently best heuristic solution.

We used a Lagrangean metaheuristic for solving the MOP, first in its static version, then extending the result to the dynamic case. As mentioned, we worked on formulation P' and relaxed constraints (2) and we considered LR as a subproblem, which can be easily solved as detailed in subsection 1.1. The upper bound provided by the LR solution can be infeasible because the resulting overlay topology could be disconnected and connections allocated to a node i have a total bandwidth greater than b_i, i.e. $\sum_{j \in N_i} x_{ij} > b_i$. The first case occurs rarely, and at rate which is compatible with the natural dynamicity of the application setting. The second case must explicitly be dealt with. In the following we will detail how to get a solution with feasible bandwidths.

Let z^*_{LR} be the solution obtained by the subgradient optimization of problem LR using penalties λ^*_i, $i \in V$. If the solution is infeasible a heuristic solution is obtained by considering the penalized costs $p'_{ij} = (p_{ij} - \lambda^*_i - \lambda^*_j)$, ranking all arcs $(i, j) \in E$ by non increasing p'_{ij} values and allocating all possible bandwidth to each successively considered connection. This approach is derived from the exact method for solving continuous knapsack problems. In our case it is not guaranteed to be optimal but it consistently produces good quality solutions in polynomial time as the highest cost operation being the ordering of the arcs.

The dynamic case has been tackled by means of a continuous application of the interwoven Lagrangean and heuristic procedures described for the static case. Moreover, the global steps required by a subgradient approach are limited to the computation of the denominator of one equation. We were able to prove [2] that substituting the summation over all nodes in the network with a summation local to each node, limited to its neighbors, does not interfere with the convergence properties of the approach. Therefore, we were able to implement our

Lagrangean heuristic in a fully distributed way.

Since variations of the network structure happen continuously, the optimization algorithm is run continuously. It is assumed that the speed of execution of iterations of the subgradient optimization procedure is higher than the rate of network changes, thus that a few subgradient iterations can be performed between consecutive network changes. The exact number of iterations is currently a parameter, named *InnerIter* in the following, which implicitly quantifies the network variability. The following algorithm is run concurrently by each node i of the network. It is assumed that each node knows its neighborhood $N_i(t)$ at each iteration t.

```
DistrLagrMOP()
Initialize penalty $\lambda_i$ and iteration counter $t$
while (true)
   do $t++$
      Solve problem $LR$
      Compute the subgradient component $g_i = \sum_{j \in N_i} x_{ij} - b_i$
      Check for infeasibility and update $\lambda_i$
      if $t$ mod $InnerIter == 0$
         then call $LagrHeuristic(p')$
              foreach $j$ in $N_i(t)$
                 do send $\lambda_i$ and $g_i$ to $j$

on receive $\lambda_j$ and $g_j$
   update $p'_{ij}$
```

```
LagrHeuristic($p'$)
initialize $s_i = b_i$
foreach $arc\ (i,j), j$ in $N_i(t)$, in nonincreasing $p'$ order
   do $slack = min\{s_i, u_{ij}\}$
      $x_{ij} = \xi_{ij} = 0$
      if $slack \geq l_{ij}$
         then $x_{ij} = slack$
              $\xi_{ij} = 1$
              $s_i = s_i - slack$
      send $x_{ij}$ and $\xi_{ij}$ to $j$

on receive $x_{ji}$ and $\xi_{ji}$
if $x_{ij} > x_{ji}$
   then $x_{ij} = x_{ji}$, $\xi_{ij} = \xi_{ji}$
```

Computational testing was performed on two problem sets, named A and B. Set A is composed by instances where nodes have a location associated, while set B is patterned following actual Internet parameters. Figure 1 (left) shows the solution for an instance of type A, where nodes correspond to peers and solid lines to connections suggested by *LagrHeuristic*. Figure 1 (right) shows one solution for an instance of type B. In this case, since the spatial position of the nodes has no meaning, we grouped the peers communicating with only another one in circles around this last. This permits an operational identification of *superpeers*.

The full presentation of our computational results, though preliminary, cannot comply with the number of page constraint. We refer the interested reader to [2] for a better account on experiments, which so far refer to runs on a single server machine simulating the distributed environment. An object implementing *DistrLagrMOP* was instantiated for every network node. We report results provided for a variable number of *DistrLagrMOP* internal loops (0 in the case of column *LagrHeu*, then 100 and 1000). Current results demonstrate the viability of the approach. The next step of our research will be to deploy the system in a full-featured distributed environment, in order to validate real-world performance.

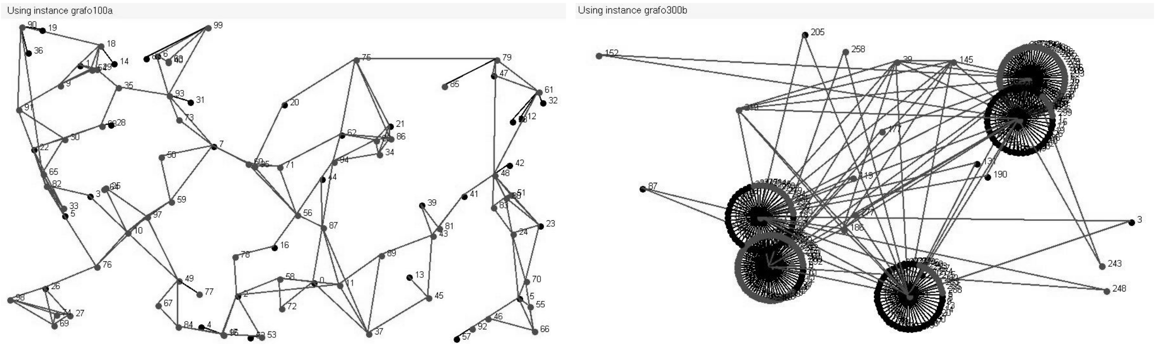

Figure 1: A lower bound solution of instances A100 (left) and B300 (right).

References

[1] Dictionary of algorithms and data structures. http://www.nist.gov/dads/, 2005.

[2] M. Boschetti, M. Jelasity, and V. Maniezzo. A local approach to membership overlay design. Working paper, Department of Computer Science, University of Bologna, 2004.

[3] P. T. Eugster, R. Guerraoui, A.-M. Kermarrec, and L. Massoulié. Epidemic information dissemination in distributed systems. *IEEE Computer*, 37(5):60–67, May 2004.

[4] M. Fisher. The lagrangean relaxation method for solving integer programming problems. *Management Science*, 27(1):1–18, 1981.

[5] F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13:533–549, 1986.

[6] F. Glover and M. Laguna. *Tabu Search.* Kluwer, Boston, 1997.

[7] M. Jelasity, A. Montresor, and O. Babaoglu. Gossip-based aggregation in large dynamic networks. *ACM Transactions on Computer Systems.* to appear.

[8] V. Maniezzo, M. Boschetti, and M. Jelasity. An ant approach to membership overlay design: Results on the dynamic global setting. In *Ant Colony Optimization and Swarm Intelligence, ANTS 2004*, volume LNCS 3172, pages 37–48. Springer, 2004.

[9] S. Voss. Meta-heuristics: The state of the art. In A. Nareyek, editor, *Local Search for Planning and Scheduling*, volume LNAI 2148, pages 1–23, Berlin, 2001. Springer Verlag.