

Provisioning Content Distribution Networks over Shared Infrastructure

Thanh Vinh Nguyen, Chun Tung Chou, Paul Boustead

Telecommunications and Information Technology Research Institute

University of Wollongong, Australia

Email: {vinh, ctchou, paul}@titr.uow.edu.au

Abstract—The current paradigm of Content Distribution Networks (CDN) deployment requires heavy infrastructure investment since a large number of servers have to be deployed over a wide area. To overcome this difficulty, we propose a new paradigm where future CDNs are to be deployed over a leased server infrastructure. This paradigm shift changes the object placement problem in current CDNs to a combined object placement and resource provisioning problem. In this paper, we present a formulation of this problem, which is a combinatorial optimization, and prove its NP-hardness. We also present a heuristic solution to this problem based on Lagrangian relaxation and subgradient optimization, which also provides error bounds so that the accuracy of the obtained solution can be evaluated. The algorithm has been implemented and tested on a number of randomly generated network topologies, and results have shown promising solution accuracy.

Index Terms—Content distribution networks, shared infrastructure, provisioning, replica placement, Lagrangian relaxation

I. INTRODUCTION

Content Distribution Networks (CDN) are networks of surrogate servers spanning the Internet, aiming to overcome network bottlenecks and improve user experience by bringing contents to network edge. In recent years, with the appearance of commercial CDNs, operated by companies like Akamai [1], Digital Island [2] and Speedera [3], this area has been receiving a great deal of attention from research community. While recognizing that Content Distribution Service is a useful service, we believe that there are still problems with the way CDNs are currently deployed. Deploying a CDN now involves building an Internet-wide network of servers to host replicated contents. As an example, Akamai currently has more than 12000 servers located in 66 countries [1]. Such scale of infrastructure is obviously challenging in financial, technical and administrative terms, both for deployment and operation of the service.

This requirement might be just too demanding an obstacle for new providers, as well as affecting commercial viability of existing ones. For this reason, especially considering the current climate of the technology market, we believe that a second Akamai phenomenon is hard to expect in a foreseeable future.

We believe that in the future, the deployment of content networks will follow one of two possible paths: 1) One is the formation of large, Internet-wide CDNs by inter-networking among regional CDN operators/carriers. This content-peering approach is being investigated by Internet Engineering Task Force content distribution internetworking working group

[4]. 2) The second alternative is to deploy CDNs using hired resources, including bandwidth, server storage space and processing power. Recent advances and research in shared network and server infrastructure such as [5], [6], [7], and references therein, has laid the vision and ground work for provisioning services using hired resources from carriers/network owners in the near future. In this paper we propose and investigate the second approach.

The “resource-hiring” approach proposes a decoupling of responsibilities between network providers and service providers, in which network providers manage the physical infrastructure and service providers lease network and server resources, in the form of overlay networks, to provision their services. We believe this environment holds the answer to the problems with today’s CDNs. It presents a less demanding initial investment and resource commitment, which facilitates deploying new CDNs, or extending current networks for new market reach. The underlying infrastructure may also be used for other applications, thus facilitating the deployment of new, innovative service concepts. Decoupling of operational responsibilities will also be a definite advantage, in which the underlying physical infrastructure is taken care of by network providers. Additionally, the shared infrastructure paradigm introduces a new dimension that will eventually transform the CDN service - the ability to adjust network resources according to needs. Our vision is a CDN that allows fine tuning resource allocations, such as opening up or closing down surrogates, when necessary.

We believe that such changes in deploying and running CDNs introduces new requirements in replicating contents and allocating resources that has yet been addressed in literature. Firstly, the question of where to place contents and how requests are directed - ie. the replica placement (RP) problem - and where and how much resources should be allocated - i.e. the provisioning problem - now become closely coupled and must be addressed simultaneously. Secondly, server resource usage, i.e. storage space and processing power, incurs charges on the CDN provider, thus a RP algorithm that fills up all potential resources in search of the best performance possible ([8], [9]) is no longer appropriate. Rather, we would expect providers to strive to achieve an acceptable performance at the least cost possible. Thirdly, the introduction of the “resource costs” concept is an important deviation that makes the problem significantly more complex. For example, depending on availability and demands, costs may be different between different locations or different types of resources (e.g. serving

power and storage). This information affects how the available resources will be used, and needs to be taken into account. The constraints on the amount of resources potentially available must also be considered, since these resources are shared by many different services.

In the paper, we have formulated a provisioning model for the CDNs over shared infrastructure, which addresses the resource allocation and object placement in parallel. The problem is formulated as a combinatorial optimization, and we have designed a solution method based on Lagrangian relaxation to solve this optimization problem.

The organization of this paper is as follows: Section II provides a brief summary of other related works, Section III introduces our problem parameters and formulation. Section IV outlines a heuristic based on Lagrangian relaxation and sub-gradient method that we propose for this problem. Section V describe our implementation of the proposed heuristic and initial results. Section VI sums up the main contribution of this work, and outlines the future directions that we will take. An appendix is also included that describes in detail the subgradient procedure used.

II. RELATED WORKS

There have been a significant amount of research in the field of replica placement algorithms for CDNs over the past few years. Some of the earliest works in the area were published in [9], [8] and [10]. In both [9] and [10], the authors aimed to find the optimal K CDN servers from a list of potential locations, so that the average latency in terms of autonomous system hops from servers to customers is minimized. The problems were formulated as classic uncapacitated K -median problems.

In [8], the author aimed to find the optimal solution as to where each content object should be replicated in an available network of servers. Although it had the same objective of minimizing the average customer-content distances, a more complex RP model was used, which takes into account individual content objects, their replicated locations, as well as space constraints at each potential site. The problem was shown to be NP-hard and was solved using heuristics.

Work in [11] examined content placement under bandwidth constraints. Assuming the knowledge of uploading bandwidth at each surrogate server, the paper aimed to find an object replication scheme so that the total access cost is minimized, given that the replicating process has to be finished within a certain time limit. Although this was an interesting scenario, the mathematical aspect was in nature the same as the space-constrained version in [8].

There are also works in other related areas that are not directly applicable but may be useful in terms of formulation and algorithms, such as proxy locations [12], cable network content replication [13], or placement of network instrumentations [14]. More recently, work in [15] provided a comprehensive

review of these and other related publications.

Interestingly, it can be seen that existing RP models share significant characteristics. Assuming unlimited server capacities seem to be a popular approach, such as those in [9], [10] or [14]. Work in [8], though did take into account the server space limits, but we believe even this does not capture the critical parameter of the problem - serving power and bandwidth limit, since this is the bottleneck that results in degraded user-perceived performance. Besides, the optimization objectives have been following a “fill-it-up” trend - looking for best performance possible using all available resources. As pointed out in Section I, such approach would not be suitable in a shared infrastructure environment. Unlike these existing works, we aim to create a model that captures all the key environment constraints. The challenge is to build suitable model that reflects important parameters, yet at the same time is manageable and solvable.

III. PROBLEM FORMULATION

A. Network and cost models

In our model we consider a graph $G = \langle V, E \rangle$, where V is the set of nodes and E is the set of edges. We define two subsets of nodes - customer nodes V_c and potential server nodes V_s , where $V = V_c \cup V_s$, $M = |V_c|$ and $N = |V_s|$. A customer node is where requests for contents are generated, which in real life may be an ISP access point. A potential server node represents a location that resources can be hired and a surrogate server can be established, e.g. a server farm. To model server capacities, each potential server $i \in V_s$ has a limit C_i on the amount of requests it can handle per unit time.

Each edge $e \in E$ has a weight that represent distances in terms of network hops between two nodes. The distances d_{ij} between any node $i \in V_s$ and $j \in V_c$ is assumed to be known and is calculated using shortest path algorithms on graph G .

A content population of K objects is used, with object k having size s_k . The amount of requests generated by customer node j for object k per unit time is represented by λ_{jk} .

A decision by the CDN owner to use potential node i for content replication and delivery is assumed to incur the following costs:

- Site start up cost γ_i - a once-off charge that models the costs involved when a new server is established (e.g. resource allocation, shipping software, configuration...)
- Storage cost - assumed to be proportional to the amount of contents stored at the site, with unit cost being α_i .
- Serving cost - intended to be a combined cost for two factors: processing power at node and delivery bandwidth, which are both linearly dependent on the amount of requests served. The cost per unit request is denoted β_i .

An “acceptable QoS threshold” is represented by a requirement of average customer-content distance for each object under T hops. The optimization objective is to find a configuration (i.e. which locations are used, which objects are replicated, and how much serving power should be used at each location) that satisfies the QoS constraint, and at the same time minimizes the total cost incurred on the CDN provider.

B. Provisioning and Object Replication Model

To formulate this optimization problem we used the following binary and continuous decision variables:

$$y_i = \begin{cases} 1 & \text{if site } i \text{ is used} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$x_{ik} = \begin{cases} 1 & \text{if object } k \text{ is replicated at site } i \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$r_{ij}^k \in [0, \lambda_{jk}] \quad (3)$$

The continuous variable r_{ij}^k is used to indicate the fraction of requests for object k from customer j that should be directed to site i .

Based on these variables, the optimization can be formulated as the following mixed integer linear programming model:

$$P = \text{Min} \underbrace{\sum_{i=1}^N \sum_{k=1}^K x_{ik} \alpha_i s_k}_{\text{Storage cost}} + \underbrace{\sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^K r_{ij}^k \beta_i}_{\text{Serving cost}} + \underbrace{\sum_{i=1}^N y_i \gamma_i}_{\text{Startup cost}} \quad (4)$$

Subject to:

$$\sum_{j=1}^M \sum_{k=1}^K r_{ij}^k \leq C_i, \forall i \quad (5a)$$

$$\sum_{i=1}^N r_{ij}^k = \lambda_{jk}, \forall j, k \quad (5b)$$

$$\frac{\sum_{i=1}^N \sum_{j=1}^M r_{ij}^k d_{ij}}{\sum_{j=1}^M \lambda_{jk}} \leq T, \forall k \quad (5c)$$

$$\sum_{i=1}^N y_i C_i \geq D_t \quad (5d)$$

$$r_{ij}^k \leq \lambda_{jk} x_{ik}, \forall i, j, k \quad (5e)$$

$$x_{ik} \leq y_i, \forall i, k \quad (5f)$$

$$x_{ik}, y_i \in \{0, 1\}; r_{ij}^k \in [0, \lambda_{jk}] \quad (5g)$$

where $D_t = \sum_{j=1}^M \sum_{k=1}^K \lambda_{jk}$ is the total demand from all customers.

In this formulation, constraint (5a) enforces the serving power limit at each potential site, (5b) means that all requests must be served, while constraint (5c) maintains the QoS threshold for each object. Constraint (5e) means that a site can only serve request for an object if that object is replicated there, and (5f) means a site must be in use for objects to be replicated there. According to constraint (5d), the total capacity of opened sites must not be smaller than the total demand. Note that this is actually a redundant constraint, which can be deduced from (5a), (5e) and (5f). Although redundant in the original model, it has been added to strengthen the Lagrangian relaxation that

will be described in Section IV. Similar techniques have often been used in literature [16].

Also note that in our model there is an underlying assumption that a load sharing system is used in the CDN, which allow requests from customers to be split and directed to a suitable server. The request splitting ratios are indicated by variable r_{ij}^k . This is necessary and useful considering that each customer node represents a customer base, such as an ISP access point, not a single end user.

We will show below that this mixed integer linear programming formulation is NP-hard, and thus we need heuristics to be able to solve problems for realistic networks. In the Section IV we will describe a proposed heuristic based on the Lagrangian relaxation technique.

C. Proving NP-hardness

We will prove the NP-hardness of this model by reducing it to a well-known NP-hard problem - the facility location problem [17], [16].

Consider our model with $K = 1$, $\gamma_i = 0, \forall i$, $T = \infty$ and denote $v_{ij} = \frac{r_{ij}}{\lambda_j}$, $v_{ij} \in [0, 1]$. Note that subscript for k has been dropped since there is only one object. In this particular case, the start up cost and performance constraint (5c) are eliminated, and the problem can be rewritten as:

$$\text{Min} \sum_{i=1}^N \sum_{j=1}^M v_{ij} \lambda_j \beta_i + \sum_{i=1}^N x_i \alpha_i s$$

Subject to:

$$\sum_{j=1}^M v_{ij} \lambda_j \leq C_i, \forall i$$

$$\sum_{i=1}^N v_{ij} = 1, \forall j$$

$$0 \leq v_{ij} \leq x_i \leq 1, \forall i, j$$

$$x_{ik} \in \{0, 1\}$$

which is a capacitated facility location problem. Thus our model is NP-hard \square

IV. SOLUTION PROCEDURE

A. Lagrangian Relaxation of the Model

Using a Lagrangian heuristic approach, we relax constraint (5e) and add the following term to the objective function:

$$\sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^K \mu_{ij}^k (r_{ij}^k - \lambda_{jk} x_{ik}), \mu_{ij}^k \geq 0$$

where μ_{ij}^k is the Lagrangian multiplier. The relaxed objective function now becomes:

$$\begin{aligned}
P_{LR} &= \sum_{i=1}^N \sum_{k=1}^K x_{ik} \alpha_i s_k + \sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^K r_{ij}^k \beta_i \\
&+ \sum_{i=1}^N y_i \gamma_i + \sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^K \mu_{ij}^k (r_{ij}^k - \lambda_{jk} x_{ik}) \\
&= \left[\sum_{i=1}^N \sum_{k=1}^K x_{ik} (\alpha_i s_k - \sum_{j=1}^M \lambda_{jk} \mu_{ij}^k) + \sum_{i=1}^N y_i \gamma_i \right] \\
&+ \left[\sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^K r_{ij}^k (\beta_i + \mu_{ij}^k) \right]
\end{aligned} \tag{6}$$

and the relaxed problem can be decomposed into two sub-problems:

$$P1(\mu) = \text{Min} \sum_{i=1}^N \sum_{k=1}^K x_{ik} (\alpha_i s_k - \sum_{j=1}^M \lambda_{jk} \mu_{ij}^k) + \sum_{i=1}^N y_i \gamma_i \tag{7}$$

Subject to:

$$\sum_{i=1}^N y_i C_i \geq D_t \tag{8a}$$

$$x_{ik} \leq y_i, \forall i, k \tag{8b}$$

$$x_{ik}, y_i \in \{0, 1\} \tag{8c}$$

and

$$P2(\mu) = \text{Min} \sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^K r_{ij}^k (\beta_i + \mu_{ij}^k) \tag{9}$$

Subject to:

$$\sum_{j=1}^M \sum_{k=1}^K r_{ij}^k \leq C_i, \forall i \tag{10a}$$

$$\sum_{i=1}^N r_{ij}^k = \lambda_{jk}, \forall j, k \tag{10b}$$

$$\frac{\sum_{i=1}^N \sum_{j=1}^M r_{ij}^k d_{ij}}{\sum_{j=1}^M \lambda_{jk}} \leq T, \forall k \tag{10c}$$

$$r_{ij}^k \in [0, \lambda_{jk}] \tag{10d}$$

With this relaxation, we have been able to separate binary variables into problem $P1$, and the continuous variable into $P2$. For each value of the Lagrangian multiplier vector μ_{ij}^k , the optimal solution to the relaxed problem can be obtained by solving $P1$ and $P2$ separately.

B. Solving the subproblems

To find the optimal solution to problem $P1$, we first leave aside constraint (8a) and further decompose the problem into

one subproblem for each location i :

$$P1_i = \text{Min} \sum_{k=1}^K x_{ik} \underbrace{(\alpha_i s_k - \sum_{j=1}^M \lambda_{jk} \mu_{ij}^k)}_{c_{ik}} + y_i \gamma_i \tag{11}$$

Subject to: (8b) and (8c).

For each of these N problems, we consider the following cases:

- 1) If $y_i = 0$ (server i closed) then:
 $x_{ik} = 0, \forall k \in 1, 2, \dots, K$, due to constraint (8b).
- 2) If $y_i = 1$ (server i opened) then:
 - for each $k \in 1, 2, \dots, K$
 - If $c_{ik} \geq 0$ then $x_{ik} = 0$
 - If $c_{ik} < 0$ then $x_{ik} = 1$
 - where c_{ik} is defined in (11).
 - $P1_i = \sum_{k=1}^K x_{ik} c_{ik} + y_i \gamma_i$

Thus server i incurs a cost of $P1_i$ if opened, and 0 otherwise.

Note that if $P1_i \leq 0$, server i is opened in the optimal solution to problem $P1$, i.e. $y_i = 1, \forall i \in V_s^- = \{i | P1_i \leq 0\}$. If $\sum_{i \in V_s^-} C_i \geq D_t$, then constraint (8a) is satisfied, and V_s^- is optimal set of servers to be opened. Otherwise (8b) can be enforced by solving the following problem, which determines which additional servers should also be opened.

$$P1 = \text{Min} \sum_{i \in V_s \setminus V_s^-} y_i P1_i \tag{12}$$

Subject to:

$$\sum_{i \in V_s \setminus V_s^-} y_i C_i \geq D_t - \sum_{i \in V_s^-} C_i \tag{13a}$$

$$y_i \in \{0, 1\} \tag{13b}$$

which is a binary knapsack problem of at most size N . There exist exact algorithms to solve this problem, for example dynamic programming algorithms [18], [19]. Note that although the addition of constraint (8a) has made solving this subproblem more complicated, our experiments showed a significant improvement in solution bounds with its inclusion.

Problem $P2$, on the other hand, is a constrained multi-commodity minimum cost flow problem. Since this is a pure linear programming problem, there exists standard algorithms that can be used. In our current implementation, the Cplex optimization package [20] is used to solve this subproblem. Note that updating the Lagrangian multiplier only modifies the objective function of this problem, thus the solution obtained in each iteration is still feasible for the problem in the next iteration. This feature has been exploited to speed up computation using Cplex advanced-basis capability.

C. Heuristic procedure

We used a subgradient method to adjust Lagrangian multiplier μ_{ij}^k and generate solution bounds. Thus instead of solving

TABLE I
NUMERICAL RESULTS.

Problem size			Heuristic result						Cplex result	
N	M	K	UB	LB	GAP (%)	c_γ (%)	c_α (%)	c_β (%)	Optimal	GAP^* (%)
10	10	10	194226	154058	26	21	43	36	189573	2.5
10	10	10	497386	359836	38	11	76	13	476175	4.5
10	10	10	363520	330475	10	81	10	9	349638	5.0
10	10	10	423032	391696	8.0	10	8	82	414730	2.0
10	11	12	11812	11250	5.0	60	15	25	11694	1.0
10	11	12	35006	30440	15	21	62	17	34726	0.8
10	11	12	47525	40620	17	15	69	16	47193	0.7
15	15	100	470689	342465	37	2	85	13	-	-
15	15	100	442080	429789	2.9	75	9	16	-	-
15	15	100	643321	585119	9.9	82	7	11	-	-
15	15	100	197202	179353	9.8	34	3.8	62	-	-
15	15	100	242183	235735	2.7	2.1	2.6	95	-	-
36	72	100	826805	787430	5.0	15	5	80	-	-
36	72	100	878436	750800	17.0	11	65	24	-	-

the original mixed integer problem, we solve the relaxed, and easier, problem for a series of different multiplier values iteratively. In each iteration, solution to the relaxed problem gives a lower bound of the original problem. We then attempt to reconstruct a feasible solution using the following procedure:

- Set $\bar{x}_{ik} = 0$ and $\bar{y}_i = 0, \forall i, k$
- Consider the solution $\{r_{ij}^k\}$ to subproblem $P2$
- For each r_{ij}^k
set \bar{x}_{ik} and \bar{y}_i to 1 if $r_{ij}^k > 0$

The set $(r_{ij}^k, \bar{x}_{ik}, \bar{y}_i)$ created by this procedure is a feasible solution to the original model, and thus gives an upper bound to the that problem. The widely-used subgradient method adjusts the Lagrangian multiplier in a way to improve the gap between these bounds, and the best feasible solution found is retained as problem solution. This procedure, with some minor modifications based on our experiments, is described in detail in the Appendix.

V. COMPUTATIONAL RESULTS

The algorithm has been implemented and tested on a number of different network topologies generated by the GT-ITM topology generator [21]. In each topology, a number of nodes are randomly selected to be potential server nodes, and the others are assumed to be customer nodes. Other parameters, including server capacities, request rates and resource costs are randomly generated. For each topology, we experimented with different cost combinations in which resource costs are varied across a wide range.

Table I shows the the results obtained from experiments with a these generated topologies. Some problems of small sizes were used so that we could obtain an exact (optimal) solution with the mixed integer programming solver provided by Cplex optimization package, which enabled a comparison between heuristic and optimal solutions.

In this table, each row show the results from a different sample problem. The first three columns are problem size parameters, namely N , M and K , the heuristic results are given in terms of upper bound UB, lower bound LB, the GAP between them, which is calculated by $GAP = \frac{UB-LB}{LB}$ and the fractional cost contributed by each type of resources – denoted c_γ , c_α , c_β for start up, storage and serving costs, respectively. Exact solutions, if obtainable, are displayed in the *Optimal* column. Column GAP^* shows the gap between this optimal result and the heuristic result.

In our experiments a maximum of 4000 iterations were used, though the program may terminate earlier if a GAP of under 2% was obtained.

Generally, these initial results indicate a gap of under 37% for all of our experiments. It has also been observed that if serving and start up costs are large compared to storage cost, the result is quite accurate (with a gap of under 10%). This gap tend to be larger (over 10%) if storage cost is set to dominate the final total cost. However, at small problem sizes, where we were able to compare this with optimal solution, it appears that even in those cases, the heuristic result is actually very close (under 5%) to the optimal solution, as shown in the GAP^* column. This suggests that although the heuristic does not always provide a good warrantee on solution quality at large storage costs, it does give a good indication of where requests should be directed, which, in turn, translates into a good feasible solution with our solution recovery procedure.

VI. CONCLUSION AND FUTURE WORK

We believe that service provisioning over shared infrastructure represents a promising future direction for the Content Distribution Network service; however, such paradigm shift implies new requirements for provisioning and replica placement algorithms that have yet to be fully addressed in current literature. In this paper we have formulated an optimization model to

address the CDN provisioning and replica placement problem that arises in this new environment. A heuristic procedure based on Lagrangian relaxation and subgradient method has also been proposed to approximate its solution. This method has been implemented and tested, and initial results have been promising in terms of solution accuracy.

One prominent disadvantage of this heuristic approach is the size of the second subproblem ($P2$). Although it is solvable, the large number of variables (NMK) will definitely affect performance for large problem sizes. In Operations Research literature, there are efficient algorithms based on column generation techniques for problems with similar structure, such as multi-commodity flow with jump constraint problems [22]. We are currently investigating the application of such algorithms to improve our heuristic. We are also working on techniques to further tighten the solution bounds, especially in the large storage cost case.

VII. APPENDIX

Subgradient procedure for updating Lagrangian multiplier:

1) *Initialize:*

Set $LB = 0$, $UB = \infty$, $t = 1$, $\mu_{ij}^{kt} = 0$

2) *At iteration t :*

Solve subproblems: $P1(\mu^t)$ and $P2(\mu^t)$

$P_{LR}^t = P1(\mu^t) + P2(\mu^t)$

Generate feasible solution based on solution r_{ij}^k of $P2$; resulted in objective function P^t

3) *Update bounds:*

$LB = \max\{LB, P_{LR}^t\}$

$UB = \min\{UB, P^t\}$

4) *Update Lagrangian multiplier:*

$$\mu_{ij}^{k(t+1)} = \max\{\mu_{ij}^{k(t)} + \theta^t(r_{ij}^k - \lambda_{jk}x_{ik}), 0\}$$

where step size θ^t is calculated by:

$$\theta^t = \frac{\phi(1.05UB - P_{LR}^t)}{\sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^K (r_{ij}^k - \lambda_{jk}x_{ik})^2} \quad (14)$$

and ϕ is a number between 0 and 2

$t = t + 1$

5) *Go back to step 2*

This procedure has some minor modifications resulted from our experimental experience:

- The gap ($UB - P_{LR}^t$) normally used in the numerator of formula (14) has been replaced with $(1.05UB - P_{LR}^t)$, as suggested in [23], which was observed to speed up convergence at the final iterations.
- The step size constant ϕ are normally initialized to 2 and halved after some iterations. In our experiments, instead

of halving ϕ , we used a graceful reduction of $\phi = 0.8\phi$ after every 50 iterations that does not improve the lower bound, thus allow the heuristic to perform more searches at large step size. This was found to improve convergence speed in the early iterations.

VIII. ACKNOWLEDGMENT

The support of the Cooperative Research Centre for Smart Internet Technology (CRC-SIT, www.smartinternet.com.au) for this work and permission to publish this paper is hereby acknowledged.

REFERENCES

- [1] Akamai. www.akamai.com.
- [2] Digital Island. www.digitalisland.net.
- [3] Speedera. www.speedera.com.
- [4] IETF Content Distribution Internetworking Working Group. www.ietf.org/html.charters/cdi-charter.html.
- [5] F. Safaei, I. Ouveysi, M. Zukerman, and R. Pattie. Carrier-scale programmable networks: Wholesaler platform and resource optimization. *IEEE Journal on Selected Areas in Communications*, 19(3), March 2001.
- [6] T. Brunner and R. Stadler. Service management in multiparty active networks. *IEEE Communications Magazine*, 38(3):144–151, 2000.
- [7] *Service Overlay Networks: SLAs, QoS and Bandwidth Provisioning*, Paris, France, November 2002.
- [8] J. Kangasharju, J. Roberts, and K. Ross. Object replication strategies in content distribution networks. In *In Proc. WCW'01: Web Caching and Content Distribution Workshop*, Boston, MA, June 2001.
- [9] L. Qiu, V. Padmanabham, and G. Voelker. On the placement of web server replicas. In *In Proc. 20th IEEE INFOCOM*, 2001.
- [10] S. Jamin, C. Jin, A. R. Kurc, D. Raz, and Y. Shavitt. Constrained mirror placement on the internet. In *Proceedings of INFOCOM*, pages 31–40, 2001.
- [11] A. Venkataramani, M. Dahlin, and P. Weidmann. Bandwidth constrained placement in a wan. *ACM Principles of Distributed Computing*, Aug 2001.
- [12] B. Li, M. J. Golin, G. F. Italiano, X. Deng, and K. Sohrawy. On the optimal placement of web proxies in the internet. In *Proceedings of INFOCOM*, 1999.
- [13] I. Cidon, S. Kutten, and R. Soffer. Optimal allocation of electronic content. In *Proceedings of INFOCOM*, pages 1773–1780, 2001.
- [14] S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang. On the placement of internet instrumentation. In *Proceedings of INFOCOM*, pages 295–304, 2000.
- [15] M. Karlsson and M. Mahalingam. Do we need replica placement algorithms in content delivery networks. In *WCW'01. IEEE International Web Content Caching and Distribution Workshop*, August 2002.
- [16] R. Sridharan. The capacitated plant location problem. *European Journal of Operational Research*, 87:203–213, 1995.
- [17] D. B. Shmoys, Éva Tardos, and K. Aardal. Approximation algorithms for facility location problems (extended abstract. pages 265–274, 1997.
- [18] S. Martello and P. Toth. *Knapsack problems*. Wiley, 1990.
- [19] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. Wiley, 1999.
- [20] ILOG Cplex Optimization Suite. www.ilog.com/products/cplex.
- [21] K. Calvert, M. Doar, and E. W. Zegura. Modeling internet topology. *IEEE Communications Magazine*, June 1997.
- [22] Jean-Francois Maurras and Yann Vaxs. Multicommodity network flow with jump constraints. *Discrete Mathematics*, 165-166(15):481–486, March 1997.
- [23] F. Glover and M. Laguna. *Modern Heuristic Techniques for Combinatorial Problems*, chapter Lagrangean Relaxation. Blackwell, Oxford, UK, 1993.