Efficient Content Distribution in Wireless P2P Networks

Qiong Sun, Victor O. K. Li, and Ka-Cheong Leung
Department of Electrical and Electronic Engineering
The University of Hong Kong
Pokfulam Road, Hong Kong, China
{joansun, vli, kcleung}@eee.hku.hk

ABSTRACT
With the development of wireless communication technologies and the popularity of the P2P applications, an important problem is to determine how to distribute data efficiently in wireless P2P networks. However, data distribution in wireless P2P networks faces many challenges compared with that in the traditional wired networks. One of the challenges is the contention problem. In this paper, we propose a conflict-free broadcasting scheme to address the problem of file sharing in wireless P2P networks. This scheme not only makes use of the broadcasting nature of the radio channel, but also schedules multiple transmissions simultaneously to take advantage of the frequency reuse of a multi-hop wireless network. It can minimize the completion time experienced by the users. We also deduce a lower bound of the completion time. Our simulation results show that our algorithm is promising.

Categories and Subject Descriptors
C.2.4 [**Cmputer-Communication Networks**]: Distributed systems - *Client/server, Distributed applications*.

General Terms
Algorithm, Design, Performance.

Keywords
Broadcasting, Content distribution, Peer-to-peer.

1. INTRODUCTION
Nowadays, peer-to-peer (P2P) file sharing applications are very popular. Since the shared files do not reside in a fixed remote server, P2P file sharing technology allows faster file transfer and thus conserves the network bandwidth. Due to the significant performance improvement with file sharing, there are many successful P2P applications such as BitTorrent [1], Gnutella [2], Kazza [3], and Napster [4].

However, they are designed for wired networks and cannot be employed directly over wireless networks due to the unique features of the wireless medium, such as the limited bandwidth and transmission range at a node, limited battery power, error-prone channels, and so on.

Due to the unique nature of wireless networks, we need to consider two more aspects in addition to those considered for the wired networks, namely, how to make full use of the network resources, and how to avoid contention during data distribution. In this paper, we employ the hop-based contention model [15]. The broadcasting nature of the wireless medium and space diversity are utilized to find as many sender and receiver pairs as possible in each distribution. Our major contributions are summarized as follows: 1) A hop-based contention graph is constructed to achieve a conflict-free schedule for data distribution among nodes. 2) The content distribution scheduling algorithm utilizing the broadcasting nature of wireless medium is developed in order to minimize the total completion time for file sharing. 3) A theoretical lower bound of the required transmission time is derived. 4) Our proposed algorithm is shown in our simulation to outperform other existing algorithms.

The rest of this paper is organized as follows. Section 2 describes the related work. An analytical model, together with the model assumptions, is presented in Section 3. Our proposed algorithm is described and analyzed in Section 4. Section 5 shows the simulation results. Future work and conclusions are summarized in Section 6.

2. RELATED WORK
There has been much work on the development of the practical P2P applications mentioned above and on academic research in improving P2P performance. Peer selection and chunk selection are two crucial factors that affect the effectiveness of the content distribution process [8]. In [12, 13, 15], peers are selected by bandwidth probing, path length estimation, and so on. Since the estimates are generally updated periodically, they may become stale and the performance suffers. However, in our

Qshine08 July 28-31, 2008, Hong Kong, China
 ISBN 978-963-9799-26-4
DOI 10.4108/ICST.QSHINE2008.3964

P2P algorithm, we do not select nodes based on the above estimates, so we do not need periodical updates.

Regarding chunk selection, Bit Torrent employs the rarest element first (REF) algorithm, in which chunks lacking in most peers are downloaded first. REF is good at increasing the availability of chunks in the network. However, the simulation results in [6] show that the performance of REF is far from optimal since REF does not take into account which peer(s) should have higher priorities as recipients according to their demands (number of chunks needed). While in most demanding node first (MDNF) algorithm, a node with the most demands will be satisfied first. In other words, REF focuses on chunk selection while MDNF deals with peer selection. Although our algorithm only accounts for chunk rarity and distributes the rarest chunks first, it enables multiple nodes to distribute simultaneously utilizing the nature of the wireless medium. The simulation results show our algorithm outperforms REF and MDNF.

The problem of broadcasting a shared content as a single chunk in heterogeneous networks has been investigated in [5, 11, 14]. It has been shown that the problem of finding an optimal broadcast schedule that minimizes the maximum completion time is NP-complete. However, these studies only analyze the problem of distributing just one chunk, whereas a content is generally divided into multiple chunks and distributed chunk-by-chunk in P2P content distribution systems. In our model, we consider how to distribute multiple chunks by finding the best broadcasting schedule.

3. PROBLEM FORMULATION

To focus on the fundamental issues, our model is based on several simplifying and assumptions: 1) All the nodes in the network are homogeneous. That is, all nodes have the same uploading and downloading bandwidths. 2) There is only a single channel for each node. This means that a node cannot transmit and receive simultaneously. In addition, a reception failure may occur due to packet collisions. 3) A file is partitioned into multiple chunks with the same size. The transmission time of one chunk between any pair of nodes is the same (say, one time slot). 4) The distribution network is quasi-static, so that network topology and channel status changes only at the boundary between any two successive time slots.

Assumption 1 and 2 will generally hold in most wireless networks. Assumption 4 will hold in those wireless networks in which the status of a network generally changes in a time scale much larger than the duration of a time slot.

3.1 Content Distribution Model

The analytical model consists of two parts. The first is a network model, which can provide connectivity and contention relationship among nodes. The other is a content processing model, which provides a schedule on how each peer distributes a chunk based on the information from the network model.

3.1.1 Network Model

Given the conditions of a wireless P2P network at time slot n, we can get the connectivity graph $G_v(n)$, and thus, build the corresponding contention graph $G_c(n)$.

Construction of connectivity and contention graph The connectivity graph $G_v(n) = (V, E)$ is bidirectional, as shown in Figure. 1. Let V be the set of nodes and E be the set of edges to indicate which pair of nodes is connected. Two nodes are considered to be connected when one node is within the transmission range of the other. $G_c(n) = (U, L)$ represents the contention graph, where U is the set of nodes and L is the set of edges to indicate which pair of nodes contend with each other. Basically, two nodes can send data in the same time slot without contention if and only if they are more than two hops away from each other. In wireless networks, nodes mainly suffer from two types of conflicts [7, 10]. The primary conflict occurs when more than one node transmit to the same destination, whereas the secondary conflict occurs when a node receiving a transmission is also within the interference range of another transmission not intended for it. Thus, two nodes within two hops in a connectivity graph $G_v(n)$ will be connected by an edge from L in its corresponding contention graph $G_c(n)$.

Broadcasting set For the efficient utilization of channels, we permit several nodes without contention to transmit within the same time slot. A set of nodes that can send in the same slot without conflict is called a broadcasting set, and there is no contention edge between any two nodes in the set in the corresponding contention graph. A maximal broadcasting set is a broadcasting set such that adding any other node to the set will result in of a contention edge between a pair of nodes in the set. *A maximum broadcasting set* is a *maximal broadcasting set* with the largest cardinality. It has been shown [14] that the problem of finding a *maximum broadcasting set* in a network is NP-complete [9]. In our paper, the *maximum weighted broadcasting set* is used to select broadcasters and distribute chunks. For a given contention graph and with weights assigned to each node, a maximum weighted broadcasting set is a maximal broadcasting set such that its weight is maximum among all maximal broadcasting sets. For example, the connectivity graph and the contention graph of a network are shown in Figures 1 and 2, respectively. The m*aximal broadcasting sets* are *{A, E}, {B}, {C, D}, and {C, E}. {A, E} and {C, D}* are *the* *maximum weighted broadcasting sets*, since the weights of both sets are three.

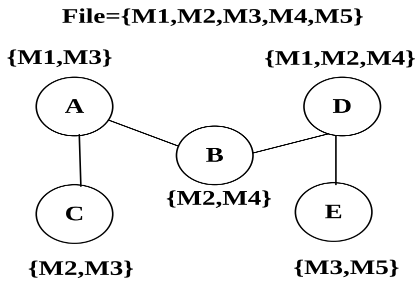

Figure 1. A connectivity graph.

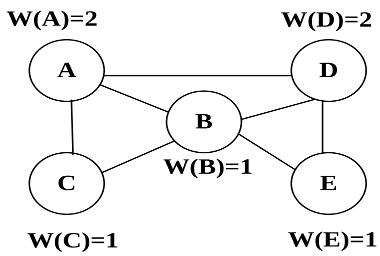

Figure 2. A contention graph.

3.1.2 Content Processing Model

Chunks are scheduled and distributed among peers based on the content possession information and peers locations in the connectivity graph.

Chunk possession vector and matrix: In a given wireless P2P network scenario, let the total number of nodes be *N*. The shared file F is divided into *M* pieces such that $F = (F_j, \forall j = 1, 2, ..., M)$. Each node possesses a subset of *F*. We use a chunk possession vector $V_{cp}(i,n)$ to represent possessions of each node *i* in time slot *n*. $V_{cp}(i,n)[j]$ is used to represent the elements in $V_{cp}(i,n)$ which is defined as:

$$V_{cp}(i,n)[j] = \begin{cases} 1, \; if \;\; node \; i \; has \; chunk \; F_j \\ 0, \; if \;\; node \; i \; does \; not \; have \; chunk \; F_j \end{cases}$$

The chunk possession information of all nodes at time slot *n* can be shown in an $N \times M$ matrix CP^n , known as the chunk possession matrix. In the example illustrated in Figure 1, *M=N=5*. Initially (*n=0*), the chunk possession matrix is:

$$CP^0 = \begin{bmatrix} 10100 \\ 01010 \\ 01100 \\ 11010 \\ 00101 \end{bmatrix}$$

At time slot 0, row i of CP^0 is the content possession vector of node i.

Chunk distribution vector: Based on the chunk possession vector from 1-hop neighbors, for each node we can construct a chunk distribution vector $V_{cd}(i,n)$, which indicates the total number of requests for each chunk possessed by node *i* by its neighbors. The element is defined

$$V_{cd}(i,n)[j] = \begin{cases} R, \; if \;\; chunk \; F_j \; possessed \; by \; node \\ \quad\; i \; is \; required \; by \; R \; neighbors \\ 0, \;\; if \;\; node \; i \; does \; not \; have \; chunk \; F_j \end{cases}$$

In Figure 1, $V_{cd}(A,n) = (2,0,1,0,0)$ means that F_1 and F_3 are requested by two and one neighbors of node A, respectively. The maximal element of $V_{cd}(i,n)$ shows which chunks of node *i* are requested the most number of neighbors. We define $C_i(n) = \max(V_{cd}(i,n)[j], \; j = 1, \; ..., M)$ to indicate which chunk has the largest value in $V_{cd}(i,n)$ and this chunk will be broadcasted by node *i*. So for node *A*, chunk 1 will be broadcasted, since $C_i(n) = \max\{V_{cd}(A,n)\} = (F_1(2))$.

4. MAX-SERVING ALGORITHM

Since we focuse on content distribution, we assume that either there is a central controller who knows the entire network topology and the requirements of each node or each node knows the global information by exchanging control messages. The aim of our algorithm is to find an optimal schedule for each time slot so as to attain the final chunk possession matrix CP^K with all its elements equal to 1 in the minimum number of time slots *K*. The algorithm is executed once at the boundary of each time slot.

4.1 MaxSer Scheduling Algorithm

Our proposed algorithm *MaxSer* is shown in the following. In this algorithm, each node is assumed to have information from its 1-hop neighbors.

MaxSer Algorithm with 1-hop Information: In time slot n, do:

1. Given a network, the 1-hop connectivity graph $G_v^1(n)$

of the network and its corresponding contention graph $G_c(n)$ can be constructed.

2. Based on the contention graph, all maximal broadcasting sets of the network can be obtained [16, 17]. For node *i*, we can get a set $S_i(n) = \{(i\ ,\ j)\ \big|\ j \text{ has no contention with } i, 1 \le j \le N\}$.
3. With the connectivity graph $G_v^1(n)$, we can get the chunk possession vector $V_{cp}(i,n)$ and the chunk distribution vector $V_{cd}(i,n)$ and $C_i(n)$ for each node i.
4. Sum each element (of set $S_i(n)$) $C_i(n)$ together and get $C_{s_i}(n)$ so that all maximal broadcasting sets can be converted into their corresponding maximal weighted broadcasting sets, and the weight of which is written as $C_{s_i}(n)$.
5. Define $W_{s_k}(n) = \max(C_{s_i}(n), i = 1, ..., N)$. S_k is the maximum weighted broadcasting set, then each node j in set S_k is selected to be broadcaster in time slot n and the corresponding chunk is sent according to $C_j(n)$. If more than one set have the same $\max(C_{s_i}(n))$, say, $W_{s_k}(n) = W_{s_l}(n) = \max(C_{s_i}(n))$, we will randomly choose one set to be the broadcaster set.
6. If each element of the chunk possession vector $V_{cp}(i,n)$ is one, the scheduling process stops; otherwise, repeat 1-6 in time slot *n+1*.

Basically, with more information, the scheduling algorithm would perform better and the completion time for content distribution can be reduced. In order to distribute more contents within the following time slots, we should consider more than 1-hop information. It is easy to achieve an optimal solution by extending *MaxSer*. We just need to collect request information of all neighbors within more hops and then apply *MaxSer*. However, it take more time to exchange messages among nodes. Thus, there is a tradeoff between better performance and more communication overheads for exchanging control information among nodes. Moreover, the topology of a wireless P2P network can be very dynamic. The connectivity relationship among nodes may vary over time. Therefore, the scheduling protocol should be designed mainly based on 1-hop network information in a dynamic environment.

4.2 Performance Analysis

In this subsection, we will analyze the lower bound of the number of time slots *K* for completing the content distribution among all nodes. In our algorithm, each node can either broadcast one chunk or receive one chunk within each time slot. For a given network connectivity graph and an initial chunk possession matrix CP^0 for all nodes participating in the network, we can analyze the lower bound *K* from two aspects: the nodes (rows of CP^0), and the chunks (columns of CP^0).

1) Nodes (rows of CP^0)

Since each node can receive at most one chunk from one of its neighbors in each time slot, the total number of time slots for each node is equal to its corresponding number of the missing chunks. The total number of *0s* across row *i* is r_i, where $r_i = \sum_{j=1}^{M}(1 - CP_{ij}^0)$. Therefore,

$$K \ge \max(r_i) \tag{1}$$

2) Chunks (columns of CP^0)

For the distribution of a certain chunk, in each time slot, the chunk cannot serve more than $D_{\max}$ nodes, where $D_{\max}$ is the maximum degree of each node. Let c_j be the total number of *1s* along column *j*. Thus, $c_j = \sum_{i=1}^{N} CP_{ij}^0$.

We can find the minimum number of *1s* along all columns $c_{\min} = \min_{j \in (1,...M)} c_j$. The chunk with $c_{\min}$ will need the most number of time slots for distribution. Q_k is defined as the number of nodes which can broadcast this chunk simultaneously in each time slot. After one time slot, there will be at most ($c_{\min} + Q_k \times D_{\max}$) 1s along this column. At time slot K, there will be $c_{\min} + Q_k \times D_{\max} \times K$ *1s* along this column, which is greater than or equal to *N*. Thus,

$$K \ge \left\lceil \frac{N - c_{\min}}{Q_k \times D_{\max}} \right\rceil \tag{2}$$

3) Combined lower bound of K

Based on the above discussion, the lower bound of the value *K* is the maximum value of the two lower bounds derived above.

$$K \ge \max(r_{\max}, \left\lceil \frac{N - c_{\min}}{Q_k \times D_{\max}} \right\rceil) \tag{3}$$

5. SIMULATION RESULTS

We randomly generate some problem instances to evaluate our algorithm. In these problem instances, we choose four representative network topologies, as shown in Figure 3. The first one is a grid topology with 9 nodes. The second one is a star topology with 10 nodes. The last two are hybrid topologies with 24 and 35 nodes, respectively. The number of chunks or chunk size is 10, 50 and 200, and the probability of any node possessing a certain chunk is 0.2 and 0.5, respectively, in the two scenarios considered. Under these different topologies and chunk possession matrices, we compare our algorithm with two classical algorithms RPF/REF and MDNF [6]. In the simulation, we focus on time slots needed for complete distribution. The simulation results show that our algorithm outperforms the others.

1) Probability of 1s in CP is 0.2

Figure 4 shows the performance comparison of the three algorithms with different chunk sizes. In all cases, the time slots needed for content distribution by *MaxSer* are smaller than the other two algorithms. The performance improvement of *MaxSer* over RPF and MDNF increases with chunk size. The reason is that *MaxSer* chooses the broadcasting set which allows more nodes to be served in each time slot, which helps distribute the content as quickly as possible.

2) Probability of 1s in CP is 0.5

Figure 5 shows the performance comparison of three algorithms by varying node size. In all cases, the time slots used by *MaxSer* are smaller than the other two algorithms. The number of time slots used by *MaxSer* does not change greatly with increasing peer size. This demonstrates that the proposed algorithm is scalable with the number of peers. In addition, we find the number of time slots with ten nodes is larger than that with 24 nodes. This is not only because there are more broadcasting sets with 24 nodes, but also the maximum degree in the 24 node graph is more than that in the ten node graph. With a given number of chunks, our proposed algorithm is effective in reducing the number of time slots needed for networks with more broadcasting sets and higher node degrees.

As shown in the simulation results, our algorithm performs better in most cases. The results also show that both the network topology and the initial distribution of chunks will greatly impact the time needed for distribution in the wireless networks.

6. CONCLUSION

In this paper, we address the content distribution problem in wireless P2P networks. We formally define the content distribution problem with the chunk possession matrix and contention graph. The broadcasters in each time slot are selected by determining the maximum weighted broadcasting set of the contention graph. Therefore, the number of time slots needed to complete content distribution is minimized and hence the overall system performance is improved. The lower bound of the number of time slots required is deduced. Our simulation results show that the proposed *MaxSer* algorithm outperforms other existing algorithms in most cases. Therefore, the proposed *MaxSer* algorithm is a promising candidate for content distribution scheduling in wireless P2P networks. In our model, to focus on the fundamental issues of content distribution, we make some assumptions such as no power constraint, homogeneous nodes, and so on. Therefore, in the future work, we would like to relax these assumptions to make it more realistic. We shall also make our model applicable to more dynamic network scenarios by incorporating mobility models of wireless nodes.

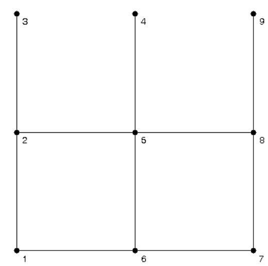

Figure 3 (a). A network topology with 9 nodes.

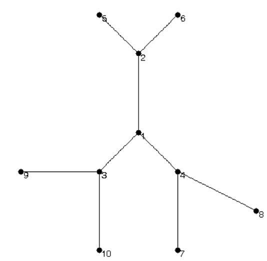

Figure 3 (b). A network topology with 10 nodes.

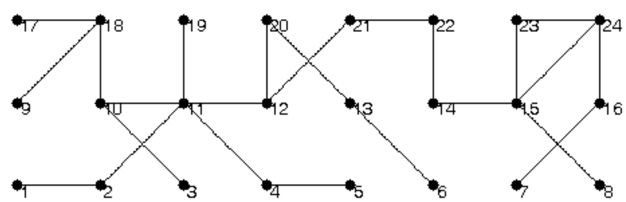

Figure 3 (c). A network topology with 24 nodes.

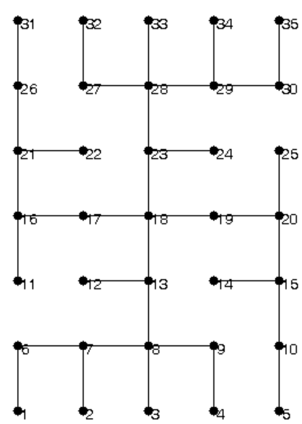

Figure 3 (d). A network topology with 35 nodes.

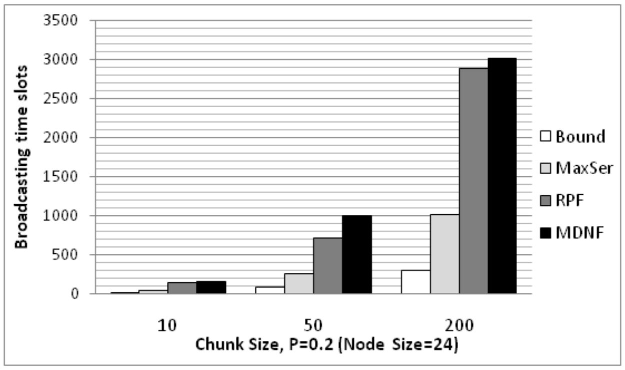

Figure 4. Number of time slots versus chunk size.

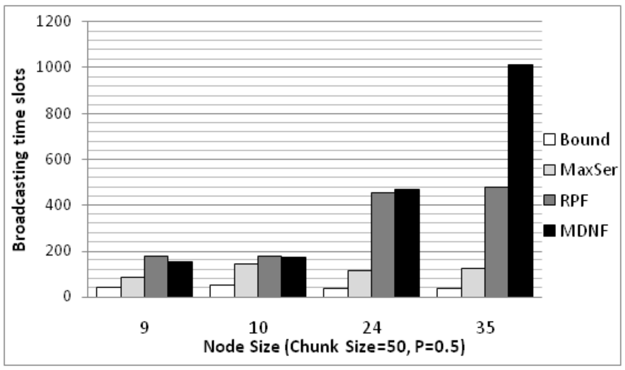

Figure 5. Number of time slots versus node size.

7. REFERENCES

[1] The official BitTorrent website, http://www.bittorrent.com/

[2] The official Gnutella website, http://www.gnutella.com/

[3] The official Kazaa website, http://www.kazaa.com/

[4] The official Nepster website, http://www.nepster.com/

[5] BHAT, P.B., RAGHAVENDRA, C.S., and PRASANNA, V.K. 2003. Efficient Collective Communication in Distributed Heterogeneous Systems. *Journal of Parallel and Distributed Computing*, Vol. 63, No. 3, pp. 251-263, March 2003.

[6] CHAN, J.S., LI, V.O.K. and LUI, K.S. 2007. Performance Comparison of Scheduling Algorithms for Peer-to-Peer Collaborative file distribution. *IEEE Journal on Selected Areas in Communications*, Vol. 25, pp. 146 – 154, January 2007.

[7] EPHREMIDS, A. and TRUONG, T.V. 1990. Scheduling Broadcasts in Multihop Radio Networks. *IEEE Transactions on Communication*. Vol. 38, pp. 456–460, April, 1990.

[8] FELBER, P. and BIERSACK, E.W. 2005. Cooperative Content Distribution: Scalability Through Self-Organization. In *Lecture Notes in Computer Science*, Vol. 3460, pp. 343-357, May 2005.

[9] GAREY, M.R. and JOHNSON, D.S. 1979. Computers and Intractibility-A Guide to the Theory of NP Completeness. W. H. Freeman.

[10] HUNG, K. and YUM, T. 1992. Fair and Efficient Transmission Scheduling in Multihop Packet Radio Networks. In *Proceedings of IEEE GLOBECOM 1992*, pp. 6–10, November 1992.

[11] KHULLER, S. and KIM, Y.A. 2004. On Broadcasting in Heterogeneous Networks. In *Proceedings of ACM-SIAM SODA '04*, pp. 1011-1020, January 2004.

[12] LEUNG, K.H. and KWOK, Y.K. 2005. An Efficient and Practical Greedy Algorithm for Server-Peer Selection in Wireless Peer-to-Peer File Sharing Networks. In *Lecture Notes in Computer Science*, Vol. 3794, pp. 1016-1025, December 2005.

[13] NG, T.S.E., CHU, Y.H.S. and ZHANG, H. 2003. Measurement- Based Optimization Techniques for Bandwidth-Demanding Peer-to-Peer Systems. In *Proceedings of IEEE INFOCOM* 2003, Vol. 3, pp. 2199-2209, 30 March – 3 April 2003.

[14] RAMASWAM, R. and PARHI, K.K. 1989. Distributed Scheduling of Broadcasts in a Radio Network. In *Proceedings of IEEE INFOCOM 1989*, Vol. 2, pp. 497-504, April 1989.

[15] SYAL, M., BREITBART, Y. and VINGRALEK, R. 1998. Selection Algorithms for Replicated Web Servers. *Reading in ACM SIGMETRICS Performance Evaluation Review*, Vol. 26, No. 3, pp. 44-50, December 1998.

[16] TSUKIYAMA, S., IDE, M., SHIRAKAWA, I. 1977. A New Algorithm for Generating All the Maximal Independent Sets. *SIAM Journal on Computing*, Vol. 6, pp. 505-517, 1977.

[17] YU, C.W. and CHEN, G.H. 1993. Generate All Maximal Independent Sets in Permutation Graphs. *International Journal of Computer Mathematics*, Vol. 47, pp. 1-8, 1993.