
R. Sai Anand and Thomas Erlebach

Routing and call control algorithms for ring networks

TIK-Report
Nr. 171, May 2003

R. Sai Anand and Thomas Erlebach
Routing and call control algorithms for ring networks
May 2003
Version 1
TIK-Report Nr. 171

Computer Engineering and Networks Laboratory,
Swiss Federal Institute of Technology (ETH) Zurich

Institut für Technische Informatik und Kommunikationsnetze,
Eidgenössische Technische Hochschule Zürich

Gloriastrasse 35, ETH Zentrum, CH-8092 Zürich, Switzerland

Routing and Call Control Algorithms for Ring Networks

R. Sai Anand T. Erlebach

Computer Engineering and Networks Laboratory
ETH Zürich

Abstract

A vast majority of communications in a network occurs between pairs of nodes, each such interaction is termed a *call*. The job of a call control algorithm is to decide which of a set of calls to accept in the network so as to maximize a certain objective, viz., the number of accepted calls or the profit associated with the accepted calls. When a call is accepted it uses up some network resources, like bandwidth, along the path through which it is routed. Thus, the call control algorithm needs to make intelligent trade-offs between resource constraints and profits. In this paper, we investigate two variants of call control problems on ring networks; in the first, the algorithm is allowed to determine the route connecting the end nodes of a call, while in the second, the route is specified as part of the input. For the first variant, we show an efficient algorithm that achieves the objective of routing and maximizing the *number* of accepted calls within an additive constant of at most 3 to an optimal algorithm. This yields a PTAS as a by-product. For the fixed path variant, we derive a 2-approximation for maximizing the *profits* (which could be arbitrary) of accepted calls. Further, for several important special cases we show polynomial time optimal algorithms or PTASes. A common feature to many of the results is the rounding of a fractional optimal solution of the relaxation of an appropriate integer linear program. While the rounding schemes themselves vary, they are similar as regards the simplicity of their details and implementation in practice. An interesting side-light, from a theoretical perspective at least, is the derivation of integral optimal solutions for the dual of the aforementioned integer linear programs.

1 Introduction

Motivation. Optical fiber based networks are increasingly replacing the traditional copper cable based ones in modern day telecommunication. They provide substantial advantages in terms of high bandwidth and capability to carry multiple types of traffic. This is in tandem with the emergence of high bandwidth applications like video-conferencing, multimedia, video on demand etc. SONET is a dominant technology standard for optical networks today. The building block of these networks, called a SONET ring, is one in which network nodes are connected together in a ring with optical fiber cables. It is therefore interesting and important to study communication problems, such as call control, that arise in ring networks.

Call admission control is a basic problem in communication networks. Within the bandwidth limitations on network links that carry data, the call control problem is to optimize the profits accrued on traffic that can be carried across the network. More concretely, the situation is the following. Network elements, like edge routers, receive a sequence of requests from nodes to establish connections with other nodes. Each such connection takes up some bandwidth along the path through which it is routed. The call control algorithm at the router needs to make a decision as to which among these requests it can accept on the network at any given time. For every request that is accepted and routed a profit is made. The trade-off between profits and bandwidth constraints will depend on the set of requests and the topology of the network.

In a typical setting, requests for connections arrive one after the other and the algorithm needs to make an on-line decision. That is, a decision to accept or reject a request cannot depend on the future but may be based on decisions made in the past. In this paper, however, we shall concentrate on the off-line problem. We shall assume that the set of requests is presented beforehand to the algorithm. There are two reasons for this. Firstly, an off-line algorithm serves as a basis for evaluating on-line ones. Secondly, there are real life scenarios like advance reservations, where the traffic requests are indeed known in advance.

A Graph Model. The call control problem can be modeled as a graph problem. The communication network is represented as a capacitated graph

$G = (V, E, w)$. The nodes and links connecting nodes are identified with the vertices and edges of the graph respectively. The bandwidths on links correspond to capacities on edges. A connection between two nodes is a path in the graph. Thus, the bandwidth constraints on the links would mean that the number of paths through an edge $e \in E$ is at most its capacity $w(e)$. It is possible to also consider a situation where different connections take up different amounts of bandwidth on the links. Here, however, we concentrate on all connections using up uniform bandwidths. If the objective is to maximize the number of connections to be accepted, in the graph we need to maximize the total number of paths. If the objective is to maximize the profit, in the graph model the sum of the profits of accepted paths should be maximized. There are two versions of call control depending on whether the call control algorithm can decide on the path it will assign to a connection or the route is pre-specified. We shall formalize these notions in the following sections.

Previous Work. Call control problems have been studied for several topologies in both the on-line and off-line settings. For chain networks, the off-line version with equal bandwidths on edges is polynomially solvable, as the problem can be modeled as a maximum k -colorable induced subgraph of interval graphs. A clever linear time implementation of this approach is presented in [7]. For the ring topology, a polynomial time optimal algorithm was given in [1], to maximize the number of accepted connections when the path for a connection is pre-specified. This result also solves the off-line problem in chains when the edge capacities are arbitrary.

When all edges have bandwidth of unity, the off-line call control problem is the well known maximum edge disjoint paths problem (MEDP). MEDP is polynomially solvable for chains [7] and undirected trees [11]. In undirected and bidirected rings, MEDP can be solved optimally as well [16, 12]. However, for bidirected trees of arbitrary degree and trees of rings, MEDP has been proved to be APX-complete [9, 8].

On-line versions of call control have also been investigated when preemption is allowed. Here, a connection once established can be removed in favour of another that is requested later on. The decision to preempt could depend on the profits of the two connections and/or their bandwidth requirements. Garay et al., in [10], study the on-line preemptive version on chains with unit edge capacities to obtain an $O(\log n)$ -competitive algorithm, where n

is the number of vertices in the chain. For chain networks, a randomized $O(1)$ -competitive algorithm is given in [2] when all edge capacities are equal. Instead of maximizing the number of accepted connections, the objective of minimizing the number of rejected connections was considered in [5]. They showed 2-competitive preemptive algorithms for chains with arbitrary capacities and for arbitrary graphs with unit edge capacities. For the off-line version, they give an $O(\log m)$ -approximation algorithm for arbitrary graphs with arbitrary capacities, where m is the number of edges in the graph.

Our Results. We study the call control problem in rings, where all connections demand unit bandwidth on the links, in two variants. In the first variant, the algorithm is allowed to determine the route of connections by itself. For this problem, we give an efficient algorithm that accepts and routes at most 3 fewer connections compared to an optimal algorithm. In the second variant, the routes for connections are predetermined and connections have arbitrary profits associated with them. We give an approximation algorithm for this case that achieves at least half the profit as compared to an optimal algorithm. Moreover, for various special cases, we provide optimal polynomial algorithms or PTASes. One of the special cases subsumes the problem considered in [1]. We provide PTASes for instances where the profit of a call is proportional to the length of its route and for the case where the length of the routes is bounded by a constant. The latter case can be solved in polynomial time using dynamic programming methods. The randomized approximation scheme is faster when the edge capacities are sufficiently high.

In Section 2, we detail the Routing and Call Admission Control (RCAC) problem. In Section 3, we present results on the Pre-routed Call Admission Control variant (PCAC). It should be remarked that the computational complexities of both these problems are as yet unresolved and are interesting open problems.

2 RCAC on Rings

Terminology: A *call* in a communication network is a pair of distinct nodes between which a connection needs to be established. It is usually specified as an (unordered) pair of the nodes, also called end points of the call. A *route* for a call is a path connecting its end points. In a well-connected network,

there might be several possible paths between two nodes. A *call set* is a set of all calls in the network which are presented to the call control algorithm for a decision on acceptance and routing.

With this terminology in place, we are now ready to define the Routing and Call Admission Control (RCAC) problem on rings:

Input Instance of RCAC: A ring (or cycle) $R = (V, E)$ on n vertices. A call set S of m calls and a capacity function $w : E \rightarrow \mathbb{Z}^+$.

In a ring, there are two possible paths connecting any two vertices. Hence, a route for a call is one of the two possible paths. A *feasible solution* to the instance is a subset $S' \subseteq S$ such that every call $\{u, v\} \in S'$ is routed and the number of routes that pass through any edge e in E is at most $w(e)$.

Objective and Output for RCAC: A feasible solution $OPT \subseteq S$ such that $|OPT|$ is the maximum possible and for every call in OPT , a route is specified.

More simply the objective is, subject to capacity constraints to route as many calls in the call set as possible. In the rest of the presentation, we abuse notation and let OPT stand for the optimal feasible set and its cardinality.

Our approach to solving the RCAC problem is to formulate it as an integer linear program and to round the optimal fractional solution of the relaxed program to a feasible solution. We shall show that the feasible solution so generated is very close to an optimal solution.

2.1 An Integer Linear Program for RCAC

The formulation of the integer linear program (ILP) for the RCAC problem is a natural one. Let the call set be $S = \{\{u_i, v_i\} : i = 1, 2, \dots, m\}$. We shall refer to a call by its index i . Further, we consider a fixed embedding of the ring on a plane and assign a clockwise direction. Let the n edges be numbered (and referred to from here on) $1, 2, \dots, n$, with edge j incident to edge $(j + 1) \bmod n$ at the former's clockwise end vertex (0 is identified with n). For each call i , introduce two indicator variables x_{i1} and x_{i2} corresponding to the two possible routes. The first of them cor-

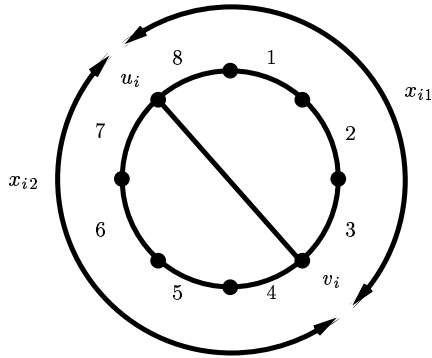


Figure 1: Ring with 8 edges. Call i and its two indicator variables

responds to the path containing edge 1 and the other to the path that does not. See Figure 1 for an illustration. For edge $j = 1, 2, \dots, n$, let $S_j = \{x_{ik} : \text{route } x_{ik} \text{ contains edge } j, i = 1, 2, \dots, m, k \in \{1, 2\}\}$. Now, the ILP looks as follows:

$$\begin{aligned}
 & \max \sum_{i=1}^m (x_{i1} + x_{i2}) \\
 & \text{subject to} \\
 & \sum_{x_{ik} \in S_j} x_{ik} \leq w(j), j = 1, 2, \dots, n \\
 & x_{i1} + x_{i2} \leq 1, i = 1, 2, \dots, m \\
 & x_{ik} \in \{0, 1\}, i = 1, 2, \dots, m, k = 1, 2
 \end{aligned}$$

Relaxing the above ILP changes the last of the constraints by admitting all fractional values between 0 and 1. The relaxed LP can be solved in time polynomial in n, m and $\log_2 w(\cdot)$. Denote the fractional optimal solution vector as $x^* = (x_{11}^*, x_{12}^*, x_{21}^*, x_{22}^*, \dots, x_{i1}^*, x_{i2}^*, \dots, x_{m1}^*, x_{m2}^*)$ and the objective value by OPT^* . It will be helpful to think of the vector x as a function on the set of routes of the calls into the real interval $[0, 1]$. Hence, we shall refer to x as a *route function* and as a $\{0, 1\}$ -*route function*, if the components of x are either 0 or 1. The import is that x_{i1}, x_{i2} are the values of $x(i1), x(i2)$ respectively. Note that for a route function, $x_{i1} + x_{i2} \leq 1, \forall$ calls i . Moreover, we shall talk about the route function independent of the edge capacities of the ring.



Figure 2: Parallel and crossing calls

2.2 Rounding Scheme

Before describing the rounding scheme it is useful to distinguish a relation between pairs of calls. Two calls $i = \{u_i, v_i\}$ and $j = \{u_j, v_j\}$ are said to be *parallel* if either their end points appear as u_i, u_j, v_j, v_i while traversing the ring in clockwise fashion or they share a common end point. Observe that since the pair of vertices in a call are unordered the order in which vertices of each call, namely u_i, v_i , themselves appear is immaterial. If two calls are not parallel then they are called *crossing*. Equivalently, a clockwise traversal encounters these end points in the order u_i, u_j, v_i, v_j . A simple observation is that when two calls are parallel one of the routes of the first call is totally contained in a route of the second and vice-versa. Parallel and crossing calls are illustrated in Figure 2.

The rounding scheme starts off by doing a preliminary set of transformations on parallel and crossing calls so that the components of the fractional optimal vector x^* are in a particular “canonical” form. It should be remarked that while we change the values of the components of x^* we do not affect either the feasibility of the resultant vector or the objective value. We proceed to describe them below.

Transformations on Parallel Calls: Let i and j be two parallel calls with the path $x_{i1}(x_{j2})$ contained in path $x_{j1}(x_{i2})$ and $x_{i1}^*, x_{i2}^*, x_{j1}^*, x_{j2}^* > 0$. The goal behind this transformation is to set at least one of the fractional values $x_{i1}^*, x_{i2}^*, x_{j1}^*, x_{j2}^*$ to zero. Let $y = \min\{x_{j1}^*, x_{i2}^*\}$. We set

$$x_{i1}^* \leftarrow x_{i1}^* + y; x_{i2}^* \leftarrow x_{i2}^* - y; x_{j1}^* \leftarrow x_{j1}^* - y; x_{j2}^* \leftarrow x_{j2}^* + y.$$

Transformations on Crossing Calls: Consider two crossing calls i and j with $x_{i1}^*, x_{i2}^*, x_{j1}^*, x_{j2}^* > 0$ and neither of $x_{i1}^* + x_{i2}^*, x_{j1}^* + x_{j2}^*$ are unity. The aim of this transformation is to either set at least one of the variables to zero or make one of the sums $x_{i1}^* + x_{i2}^*, x_{j1}^* + x_{j2}^*$ equal unity. This is achieved in a slightly more involved transformation shown below:

Set $\epsilon_i = 1 - (x_{i1}^* + x_{i2}^*); \epsilon_j = 1 - (x_{j1}^* + x_{j2}^*)$
and $y = \min\{\frac{\epsilon_i}{2}, \frac{\epsilon_j}{2}, x_{i1}^*, x_{i2}^*, x_{j1}^*, x_{j2}^*\}$.

Case 1: if $y = \frac{\epsilon_i}{2}$,
 $x_{i1}^* \leftarrow x_{i1}^* + y; x_{i2}^* \leftarrow x_{i2}^* + y; x_{j1}^* \leftarrow x_{j1}^* - y; x_{j2}^* \leftarrow x_{j2}^* - y$
 $\implies x_{i1}^* + x_{i2}^* = 1$.

Case 2: if $y = \frac{\epsilon_j}{2}$,
 $x_{i1}^* \leftarrow x_{i1}^* - y; x_{i2}^* \leftarrow x_{i2}^* - y; x_{j1}^* \leftarrow x_{j1}^* + y; x_{j2}^* \leftarrow x_{j2}^* + y$
 $\implies x_{j1}^* + x_{j2}^* = 1$.

Case 3: if $y = x_{i1}^*$ or x_{i2}^* ,
 $x_{i1}^* \leftarrow x_{i1}^* - y; x_{i2}^* \leftarrow x_{i2}^* - y; x_{j1}^* \leftarrow x_{j1}^* + y; x_{j2}^* \leftarrow x_{j2}^* + y$
 $\implies x_{i1}^* = 0$ or $x_{i2}^* = 0$.

Case 4: if $y = x_{j1}^*$ or x_{j2}^* ,
 $x_{i1}^* \leftarrow x_{i1}^* + y; x_{i2}^* \leftarrow x_{i2}^* + y; x_{j1}^* \leftarrow x_{j1}^* - y; x_{j2}^* \leftarrow x_{j2}^* - y$
 $\implies x_{j1}^* = 0$ or $x_{j2}^* = 0$.

If x^* is a basic feasible solution to the linear program it can be proved that the x^* values of pairs of crossing calls are already in the canonical form that we require. Indeed, consider a pair of crossing calls i and j that have x^* values not in the canonical form. Compute two new feasible solutions from x^* as follows (we use $y(\neq 0)$ computed in the description above): (i) add (subtract) $y/2$ to (from) both components of x^* values of call i (j), the other components of x^* remain the same and (ii) subtract (add) $y/2$ from (to) both components of x^* values of call i (j), the other components of x^* remain the same. Clearly, the original x^* can be expressed as a convex combination of these two new vectors. Thus, if the x^* values of i and j were not in canonical form it is not basic. Conversely, if x^* were basic then for the crossing calls i and j , the corresponding values are in the canonical form. For parallel calls,

however, a basic solution need not be in the canonical form. Consider a ring on 4 vertices with edge capacities 1 and two parallel calls i and j in this ring. The feasible solution $x_{i1} = x_{i2} = x_{j1} = x_{j2} = 0.5$ is basic but is not in the canonical form we require for parallel calls.

In any case, the above described transformations convert any feasible solution of the LP into the canonical form we want. These transformations performed on every pair of calls partitions the call set into four categories according to the values of their corresponding indicator variables in the optimal solution vector x^* :

- A) Calls for which both the corresponding indicator variables are set to zero. Let the set be denoted by $S_{(a)}$ and the sum of their x^* values by $x^*(S_{(a)}) = 0$.
- B) Calls for which exactly one of the corresponding indicator variables is non-zero. Let the set be denoted by $S_{(b)}$ and the sum of their x^* values by $x^*(S_{(b)})$.
- C) Calls which are pairwise crossing but the sum of their (non-zero) indicator variables equals unity. Let the set be denoted by $S_{(c)}$ and the sum of their x^* values by $x^*(S_{(c)})$.
- D) At most one call for which the sum of the (non-zero) indicator variables is less than one. Let the call be D and the sum of its x^* values be $x_{D1}^* + x_{D2}^* < 1$ with $0.5 > x_{D2}^* \leq x_{D1}^*$, say.

Clearly, the rounding scheme needs to handle the class B, C and D calls by setting the indicator variables to either 0 or 1. The different categories of calls require different rounding mechanisms. We shall describe these rounding schemes in a sequence of lemmas below. These lemmas prove that the solution obtained through the rounding both maintains feasibility and remains “close” to an optimal solution at the same time. While the calls are partitioned into different classes based on the route function x^* values, for ease of presentation a set of calls will be said to be of a particular class if their x^* values satisfy the conditions for that class.

2.2.1 Rounding of Class B Calls

Since calls in class B have one of their two indicator variables set to zero, the route function x can be restricted to be defined on the unique route for each call that received a non-zero value. Instead of calls, we need only concentrate on the unique path for each call in class B. Accordingly, we show

the rounding on a set of paths.

Lemma 2.1 (Rounding on the line) *Let S be a set of paths on a line $L = (V, E)$ with capacity function $w : E \rightarrow \mathbb{Z}_0^+$ on the edges. Let $x : S \rightarrow [0, 1]$ be a function that assigns fractional values to the set of paths and $x(S) = \sum_{s \in S} x(s)$. Further, let $x(\cdot)$ be such that the sum of x values of paths through an edge e on the line is at most $w(e)$. Then \exists a function $x' : S \rightarrow \{0, 1\}$ such that $\forall e, \sum_{s \in S: s \text{ contains } e} x'(s) \leq w(e)$ and $x'(S) = \sum_{s \in S} x'(s) \geq \lceil x(S) \rceil$.*

Proof: Let $S = \{s_1, s_2, \dots, s_m\}$ be a set of m paths on the line such that the right end point of s_{i+1} lies to the right (possibly even coinciding) of the right end point of $s_i, i = 1, 2, \dots, m - 1$. We prove the lemma by induction on the cardinality of S .

Basis Step: If $m = 1$, then define x' as:

$$x'(s_1) = \begin{cases} 0, & \text{if } x(s_1) = 0 \\ 1, & \text{if } x(s_1) > 0 \end{cases}$$

Induction Step: Assume the lemma holds true for S with cardinality less than $m (> 1)$. If $x(s_1) = 0$ or 1 , we can invoke the induction hypothesis on $S \setminus \{s_1\}$ and prove the lemma for S . Assume $0 < x(s_1) < 1$. We define $x'(\cdot)$ as follows. Set $x'(s_1) = 1$ and $\epsilon = 1 - x(s_1)$. If $x'(s_1)$ together with $x(s_2), x(s_3), \dots, x(s_m)$ does not violate any of the capacities of edges in s_1 , then this is similar to the case $x(s_1) = 1$. Otherwise, let e be the leftmost edge on s_1 that was violated by rounding $x'(s_1)$ to 1 . Let $s_{e_1}, s_{e_2}, \dots, s_{e_k}$ be the set of paths that pass through e and appearing in increasing order of their right end points. If $\sum_{j=1}^k x(s_{e_j}) \geq \epsilon$ then let e_l be the least index such that $\sum_{j=1}^l x(s_{e_j}) \geq \epsilon$. Reassign $x(s_{e_j}) = 0, j = 1, 2, \dots, l - 1$ and $x(s_{e_l}) = \sum_{j=1}^l x_{old}(s_{e_j}) - \epsilon$ ($x_{old}(\cdot)$ denotes the $x(\cdot)$ values before the reassignment). Now, $x'(s_1)$ together with the new $x(s_{e_1}), x(s_{e_2}), \dots, x(s_{e_k})$ add up to at most $w(e)$. If $\sum_{j=1}^k x(s_{e_j}) < \epsilon$, reassign $x(s_{e_j}) = 0, j = 1, 2, \dots, k$. Edge e is not violated anymore and edges to the right of e and in s_1 can have an excess of at most $\epsilon - \sum_{j=1}^k x_{old}(s_{e_j})$. We can repeat the above procedure for the next leftmost violated edge after e and continue till no edge in s_1 is violated. Observe that the new x values are such that $x(S \setminus \{s_1\}) \geq x_{old}(S \setminus \{s_1\}) - \epsilon$.

Further, the new x values of paths through an edge e contained in path s_1 sum up to at most $w(e) - 1$. We can now apply the induction hypothesis on the set $S \setminus \{s_1\}$ with the new x values to obtain a function $x' : S \setminus \{s_1\} \rightarrow \{0, 1\}$ that satisfies the conditions of the lemma. This x' together with $x'(s_1) = 1$ is the required x' for S , since at any edge e the sum of the x' values of paths through e add up to at most $w(e)$ and,

$$\begin{aligned}
x'(S) &= x'(s_1) + x'(S \setminus \{s_1\}) \\
&\geq 1 + \lceil x_{old}(S \setminus \{s_1\}) - \epsilon \rceil \\
&= \lceil x_{old}(S \setminus \{s_1\}) - \epsilon + 1 \rceil \\
&= \lceil x_{old}(S) \rceil, & x_{old}(s_1) = 1 - \epsilon
\end{aligned}$$

Hence, the lemma. □

This rounding lemma for the line serves as a starting step to round the values for paths on the ring. The next lemma captures this.

Lemma 2.2 (Rounding on the ring) *Let S be a set of paths on a ring $R = (V, E)$ with capacity function $w : E \rightarrow \mathbb{Z}_0^+$ on the edges. Let $x : S \rightarrow [0, 1]$ be a function that assigns fractional values to the set of paths and $x(S) = \sum_{s \in S} x(s)$. Further, let $x(\cdot)$ be such that the sum of x values of paths through an edge e on the ring is at most $w(e)$ and for some edge e_{sat} the sum is exactly $w(e_{sat})$. Then \exists a function $x' : S \rightarrow \{0, 1\}$ such that $\forall e, \sum_{s \in S: s \text{ contains } e} x'(s) \leq w(e)$ and $x'(S) = \sum_{s \in S} x'(s) \geq \lceil x(S) \rceil - 1$.*

Proof: Consider the edge e_{sat} in the ring and the set of paths $S_{e_{sat}} \subseteq S$ that pass through it. If there were two paths s_{e_1}, s_{e_2} through e_{sat} such that the former is contained in the latter then consider the following reassignment of their x values; $x(s_{e_1}) \leftarrow \min\{1, x(s_{e_1}) + x(s_{e_2})\}$, $x(s_{e_2}) \leftarrow x(s_{e_2}) + x_{old}(s_{e_1}) - x(s_{e_1})$, where $x_{old}(s_{e_1})$ is the value of $x(s_{e_1})$ before the reassignment. Obviously, with this reassignment the new x values add up to their old x values. It is also easy to observe that this reassignment either makes the x value of the shorter path 1 or the x value of the longer path 0. Thus, w.l.o.g. we can assume that all paths through e_{sat} which have x values in $(0, 1)$ are not fully contained in each other. Call these paths $S_{e_{sat}}^{(1)} = \{s_1, s_2, \dots, s_k\}$ where the order in which they appear is according to the increasing order of their clockwise end points. Let

e_j be the smallest index such that $\sum_{i=1}^{e_j} x(s_i) \geq j, j = 1, 2, \dots, x(S_{e_{sat}}^{(1)}) - 1$. Define $x'(s_{e_j}) = 1, j = 1, 2, \dots, x(S_{e_{sat}}^{(1)}) - 1$ and $x'(s_i) = 0$, for $s_i \in S_{e_{sat}}^{(1)} \setminus \{s_{e_1}, s_{e_2}, \dots, s_{e_{x(S_{e_{sat}}^{(1)})-1}}\}$. Also, set $x'(s) = x(s)$, for $s \in S_{e_{sat}} \setminus S_{e_{sat}}^{(1)}$. Recall that for the paths in $S_{e_{sat}} \setminus S_{e_{sat}}^{(1)}$ the x values are either 0 or 1.

Argument: For any edge e' in the ring, the sum of the x' values of paths in $S_{e_{sat}}^{(1)}$ that pass through it is at most the sum of x values rounded down.

Proof: Consider an edge e' in the ring.

Case 1: s_1 and s_k pass through e' .

If all paths in $S_{e_{sat}}^{(1)}$ pass through e' then the sum of their x' values is $x(S_{e_{sat}}^{(1)}) - 1$ which is one less than the sum of their x values, $x(S_{e_{sat}}^{(1)})$ as required. Otherwise, because paths in $S_{e_{sat}}^{(1)}$ are not contained in each other, it should be the case that paths $s_i, s_{i+1}, \dots, s_{i+l}$, for some $i, l, 2 \leq i \leq i+l \leq k-1$, do not pass through e' . The x values at e' then would be $\sum_{j=1}^{i-1} x(s_j) + \sum_{j=i+l+1}^k x(s_j)$ and that of x' would be at most $\lfloor \sum_{j=1}^{i-1} x(s_j) \rfloor + \lfloor \sum_{j=i+l+1}^k x(s_j) \rfloor$ which satisfies the conditions of the argument.

Case 2: exactly one of s_1 and s_k contains e'

Say, s_1 contains e' , the other case being symmetric. Again by non-containment property, there is an $i, 1 \leq i \leq k-1$, such that all paths with index from 1 through i pass through e' . The sum of the x values of these paths is $\sum_{j=1}^i x(s_j)$ which is at least $\lfloor \sum_{j=1}^i x(s_j) \rfloor = \sum_{j=1}^i x'(s_j)$. This completes the proof of the argument.

□

Now, consider all paths that do not pass through e_{sat} , they lie on the line obtained by removing the edge e_{sat} from the ring. Therefore, we can invoke Lemma 2.1, to obtain a x' function on them which satisfies the condition that the sum of x' values passing through any edge is at most the rounded up value of the sum of their x values. This, combined with the statement of the above argument implies that the x' values of paths in S that pass through

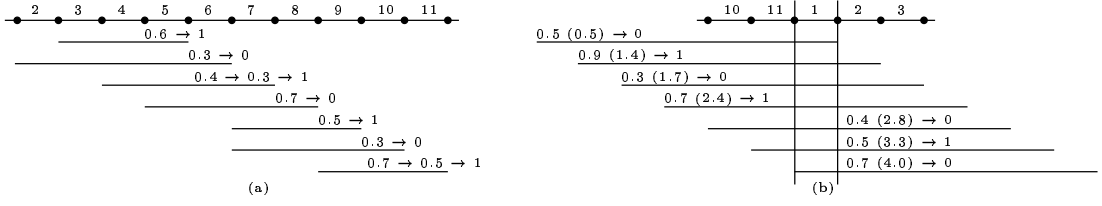


Figure 3: Rounding of Class B calls. (a) shows rounding of routes not through edge 1. (b) shows rounding of routes through edge 1. Figure does not show routes which received x^* values 0 or 1.

any edge e of the ring sum up to at most the capacity of that edge, $w(e_{sat})$. Further, we have

$$\begin{aligned}
x'(S) &= x'(S_{e_{sat}}) + x'(S \setminus S_{e_{sat}}) \\
&\geq x(S_{e_{sat}} \setminus S_{e_{sat}}^{(1)}) + x(S_{e_{sat}}^{(1)}) - 1 + \lceil x(S \setminus S_{e_{sat}}) \rceil, \text{ from Lemma 2.1} \\
&= x(S_{e_{sat}}) - 1 + \lceil x(S) - x(S_{e_{sat}}) \rceil \\
&= \lceil x(S) - x(S_{e_{sat}}) + x(S_{e_{sat}}) \rceil - 1, \quad x(S_{e_{sat}}) = w(e_{sat}) \text{ is an integer} \\
&= \lceil x(S) \rceil - 1.
\end{aligned}$$

□

Lemma 2.2 immediately suggests a rounding scheme for class B calls such that rounded values at any edge sum up to at most the rounded up value of the sum of their x^* values and at the same time lose at most one from their cumulative sum. We note that if none of the x^* values at an edge sum exactly to the rounded up value at an edge then we can increase at least one of the x^* values to satisfy the condition or make all x^* values equal 1. This is summarized in the corollary below.

Corollary 2.3 (Rounding class B calls) *Given a set of class B calls $S_{(b)}$ with a corresponding route function x^* . There exists a $\{0, 1\}$ -route function x' such that*

(i) *at every edge the sum of the x' values through it is at most the rounded up value of the sum of the x^* values.*

(ii) $x'(S_{(b)}) = \sum_{i \in S_{(b)}} (x'_{i1} + x'_{i2}) \geq \lceil x^*(S_{(b)}) \rceil - 1$.

Figure 3 shows an example of rounding of class B calls.

2.2.2 Rounding of Class C Calls

Our next step is to describe a rounding for the class C calls. The general idea behind the rounding is that we can reassign the x^* values corresponding to a call to be either 0, 0.5 or 1 without losing on their contribution to the objective value or feasibility. These x^* values can then be rounded to 0 or 1. However, to maintain feasibility we will need to throw away a constant number of calls, bounded by 2 from above. We start with a lemma that does the rounding when the two variables corresponding to a call are exactly 0.5 each. A definition is in order before we state the lemma, two edges in a ring are said to be *diametrically opposite* if the set of vertices on which they are incident is exactly the set of the end points of two crossing calls. We assume that every vertex in the ring is an end point of a call. If not, we could merge such a vertex with one of its neighbours without losing generality.

Lemma 2.4 (Rounding an even number of pairwise crossing calls)

Given a set of $2m$ mutually crossing calls in a ring and a route function x such that $x_{i1} = x_{i2} = 0.5, i = 1, 2, \dots, 2m$. There exists a $\{0, 1\}$ -route function x' and a call $j, 1 \leq j \leq 2m$ such that

(i) $x'_{j1} = x'_{j2} = 0,$

(ii) $x'_{i1} + x'_{i2} = 1, i \neq j,$

and (iii) the sum of the x' values at any edge is at most the sum of the x values.

Further to the point, the sums of the x' values at diametrically opposite edges in the ring are exactly m and $m - 1$, respectively.

Proof: Proof by induction on m . For $m = 1$, there are exactly two crossing calls in the ring, say 1 and 2. Assign $x'_{11} = 1$ and $x'_{12} = x'_{21} = x'_{22} = 0$. It is easy to verify the conditions of the lemma for this function x' .

Let the lemma hold for $m - 1$ (≥ 1), we shall prove the statement holds for m . Let $2m$ pairwise calls be given. W.l.o.g. we can assume that every vertex in the ring is an end point of one of the calls. Remove two of these calls which have end points at edge 1 in the ring, call them i and j . Now, we have $2m - 2$ calls for which the lemma holds by inductive hypothesis. Thus, there is one call among these $2m - 2$ calls which has been assigned 0 for both its paths and the rest of them received 1 for one of their paths and 0 for the other. Further, the sum of the x' values at an edge was $m - 1$ or

$m - 2$. If edge 1 received a sum of $m - 1$ then assign $x'_{i1} = x'_{j1} = 0$ and $x'_{i2} = x'_{j2} = 1$. This makes the sum of x' values at all edges other than 1 and its diametrically opposite edge go up by exactly one. And, for edge 1 the sum of the x' value remains at $m - 1$ while for the diametrically opposite edge of 1 the sum of the x' values goes up by 2 to m . The case when edge 1 received a sum of x' values of $m - 2$ is complementary. But, these are exactly the conditions of the lemma for a set of $2m$ calls, which concludes the induction argument and the proof of the lemma. □

For the sake of presentation, we introduce a fictitious call with index 0, which refers to none of the calls in the input instance.

Corollary 2.5 (Rounding pairwise crossing calls) *Given a set of m mutually crossing calls in a ring and a route function x such that $x_{i1} = x_{i2} = 0.5, i = 1, 2, \dots, m$. There exists a $\{0, 1\}$ -route function x' and a call*

$j, 0 \leq j \leq m$ such that

(i) $x'_{j1} = x'_{j2} = 0$,

(ii) $x'_{i1} + x'_{i2} = 1, i \neq j$,

and (iii) the sum of the x' values at any edge is at most the sum of the x values rounded up.

Also, $\sum_{i=1}^m (x'_{i1} + x'_{i2}) = \begin{cases} \sum_{i=1}^m (x_{i1} + x_{i2}) - 1 & \text{for even } m \\ \sum_{i=1}^m (x_{i1} + x_{i2}) & \text{for odd } m \end{cases}$

Proof: If m is even, the statement directly follows from Lemma 2.4. If m is odd, introduce a new call 0 by splitting a pair of diametrically opposite edges into two. The new vertices are the end vertices of call 0. It is clear that this new call crosses all the other calls. Assign $x_{01} = x_{02} = 0.5$. The sum of the x values at any edge is $(m + 1)/2$, the rounded up sum of the x values of the m calls. Now, invoking the previous lemma the x' values of one of the $(m + 1)$ calls is set to zero. By symmetry, we can set $x'_{01} = x'_{02} = 0$. Thus, for the odd case, $j = 0$. Now, $x'_{i1} + x'_{i2} = x_{i1} + x_{i2}, \forall i \neq j$. Hence,

$$\begin{aligned} \sum_{i=1}^m (x'_{i1} + x'_{i2}) &= \sum_{i, i \notin \{0, j\}} (x_{i1} + x_{i2}) \\ &= \sum_{i=1}^m (x_{i1} + x_{i2}) - \sum_{i=j \neq 0} (x_{i1} + x_{i2}) \\ &= \begin{cases} \sum_{i=1}^m (x_{i1} + x_{i2}) - 1 & \text{if } m \text{ even, since } j \neq 0 \\ \sum_{i=1}^m (x_{i1} + x_{i2}) & \text{if } m \text{ odd, since } j = 0 \end{cases} \end{aligned}$$

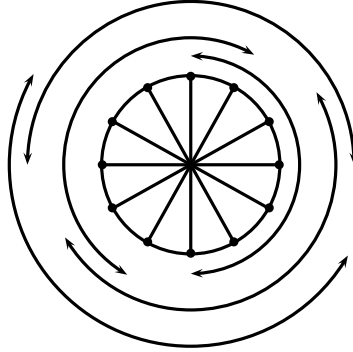


Figure 4: Rounding of class C calls. There are 6 mutually crossing calls corresponding to the chords. The rounding scheme routes 5 of them.

Also, it follows that the sum of the x' values at an edge is at most $\lceil m/2 \rceil$, which is the sum of the x values at the edge rounded up.

□

Refer to Figure 4 for how the mutually crossing calls with $x_{i_1} = x_{i_2} = 0.5$ are routed by the rounding scheme.

Recall that class C calls had their corresponding x^* values summing to exactly one. We have just shown that if these x^* values are 0.5 each then there exists a rounding that loses at most 1 call compared to an optimal solution. The next step is to show how to achieve half integral values from arbitrary ones. First, we will discard one of the calls from the set of crossing calls. Next, for the remaining calls we appeal to the powerful Okamura-Seymour theorem in [13] to get half integer values.

Theorem 2.6 (Okamura-Seymour Theorem) *If $G = (V, E)$ is a planar graph with edge capacities $w(e), e \in E$ and can be drawn such that vertices $s_1, s_2, \dots, s_k, t_1, t_2, \dots, t_k$ are all on the boundary of the infinite region, then the following are equivalent:*

(i) *For $1 \leq i \leq k$ there is a flow F_i from s_i to t_i of value q_i such that for each edge $e \in E$*

$$\sum_{i=1}^k |F_i(e)| \leq w(e).$$

(ii) For each $X \subseteq V$,

$$\sum_{e \in \partial(X)} w(e) \geq \sum_{i \in D(X)} q_i$$

($\partial(X) \subseteq E$ is the set of edges with one end in X and the other in $V \setminus X$. $D(X) \subseteq \{1, 2, \dots, k\}$ is $\{i : 1 \leq i \leq k, \{s_i, t_i\} \cap X \neq \emptyset \neq \{s_i, t_i\} \cap (V \setminus X)\}$.) Furthermore, if q and w are integer valued, then the flows F_i may be chosen half-integer valued.

The relation between the theorem and our problem is readily apparent. The ring is a planar graph and all the vertices in it indeed lie on the outer infinite face. The flows correspond to the paths connecting the end vertices. Thus, if we are able to show that the mutually crossing C calls satisfy condition (ii) of the theorem then we can obtain half-integer valued flows (or equivalently, half-integer values for the routes of a call). Lemma 2.9 addresses this. But, first we need to identify one call among the class C calls which will be discarded for the above theorem to be applied.

We start with some more terminology. Given a ring on $2m$ vertices, two edges are *almost diametrically opposite* if they have $m - 2$ edges between them. Note that between any pair of diametrically opposite edges there are exactly $m - 1$ edges. For every edge there is exactly one diametrically opposite edge and there are two almost diametrically opposite edges. For a set of m mutually crossing calls with a route function x with $x_{i1} + x_{i2} = 1, \forall i$, the total of the rounded down sums of x values at diametrically opposite edges is at least $m - 1$ and for almost diametrically opposite edges is at least $m - 2$.

Lemma 2.7 *Given a set of m mutually crossing calls with a route function x such that $x_{i1} + x_{i2} = 1, \forall i$ and $x_{i1}, x_{i2} \notin \{0, 1\}$ and an edge e_0 such that the total of the rounded down sum of x values through it and the rounded down sum of x values through its almost diametrically opposite edge is $m - 2$. There exist two consecutive edges in the ring such that the rounded down sums of the x values through them are equal.*

Proof: Assume to the contrary that for every pair of consecutive edges the rounded down values are unequal. Let the sum of the x values at an edge e be $x(e)$. For two consecutive edges e, e' it is true that $|x(e) - x(e')| < 1$. Therefore, $\lfloor x(e) \rfloor - \lfloor x(e') \rfloor = \pm 1$. Consider the edge e_0 and one of its almost diametrically opposite edges e_{m-1} in the ring. They have $m-2$ edges between them (traversing the ring in one of the two possible ways). Denote the edge that has exactly $k-1$ edges between it and e_0 in this above traversal by $e_k, k = \{1, 2, \dots, m-1\}$. It can be proved that $\lfloor x(e_0) \rfloor - \lfloor x(e_k) \rfloor \in \{\pm k, \pm(k-2), \pm(k-4), \dots, \pm(k-2\lfloor \frac{k}{2} \rfloor)\}$. Indeed for $k=1$, it is trivially true. For $k \geq 2$, $\lfloor x(e_0) \rfloor - \lfloor x(e_k) \rfloor \in \{\lfloor x(e_0) \rfloor - \lfloor x(e_{k-1}) \rfloor \pm 1\}$, since $\lfloor x(e_{k-1}) \rfloor - \lfloor x(e_k) \rfloor = \pm 1$. From here, the above statement follows. We now have $\lfloor x(e_0) \rfloor - \lfloor x(e_{m-1}) \rfloor \in \{\pm(m-1), \pm(m-3), \dots, \pm(m-2\lceil \frac{m}{2} \rceil - 1)\}$. But, $\lfloor x(e_0) \rfloor + \lfloor x(e_{m-1}) \rfloor = m-2$, implying $\lfloor x(e_0) \rfloor - \lfloor x(e_{m-1}) \rfloor = m-2-2\lfloor x(e_{m-1}) \rfloor$. Or $\lfloor x(e_0) \rfloor - \lfloor x(e_{m-1}) \rfloor \in \{\pm(m-2), \pm(m-4), \dots, \pm(m-2\lfloor \frac{m}{2} \rfloor)\}$. A contradiction. Thus our hypothesis that no two successive edges have equal rounded down sum of x values is impossible, proving the claim. □

Corollary 2.8 *Given a set of m mutually crossing calls together with a route function x such that $x_{i_1} + x_{i_2} = 1, \forall i$. There exist two consecutive edges such that the total of the rounded down sums of x values at each edge and the diametrically opposite edge of the other consecutive edge is at least $m-1$. (Note that the diametrically opposite edge of an edge is an almost diametrically opposite edge of its consecutive edge.)*

Proof: If no pair of consecutive edges satisfying the conditions stated in the corollary exists, we can arrive at a contradiction. It follows from this assumption that there exists a pair of almost diametrically opposite edges whose total of the rounded down sums of x values is $m-2$. By Lemma 2.7, there are two consecutive edges e_1, e_2 such that their rounded down sums of the x values are equal. But, the rounded down sums of the x values of diametrically opposite edges add up to at least $m-1$. Let e'_1, e'_2 be diametrically opposite edges of e_1, e_2 , respectively. For two consecutive edges, one's diametrically opposite edge is the other's almost diametrically opposite edge. Thus, the rounded down sums of x values of $e_1(e_2)$ and its almost diametrically opposite edge $e'_2(e'_1)$ add up to at least $m-1$. This contradicts our supposition.

□

Consider any two edges e_l, e_r in the ring. We shall use the edges e_1, e'_1, e_2, e'_2 identified in the above proof. Let the call, one end vertex of which is incident on edges e_1, e_2 , be k . A call is said to *cross* a set of edges in the ring if its end vertices lie in two different components of the graph obtained after removal of the set of edges from the ring. If call k crosses the cut formed by e_l, e_r , then the total of the rounded down sums of x values is at least the number of calls crossing them excluding call k . To see this, observe that every call that crosses the cut e_l, e_r contributes a value of 1 to the sum of x values on these edges. Thus, the sum of the x values on the cut e_l, e_r due to the crossing calls is an integer. Rounding the x sums down would decrease the total by one. Hence, the rounded down sums of x values at e_l, e_r would be at least the number of crossing calls minus one. On the other hand, if they lie on the same side of call k the above sum is at least the number of calls crossing them. Indeed, the total of the rounded down sums of x values at e_1 and e'_2 is $m - 1$. W.l.o.g assume that all four edges, e_l, e_r, e_1, e'_2 lie on the same side of call k and appear in that order on a clockwise traversal of the edges. The difference in rounded down sum of x values at consecutive edges is at most 1. Thus, if there are t vertices between e_r and e_1 and s vertices between e_l and e'_2 , the total of the rounded down sums of the x values at e_l and e_r is at least $m - 1 - s - t$. But this value is exactly the number of calls crossing e_l and e_r . Thus, we can conclude that after removing call k from the set of mutually crossing calls, for any pair of edges in the ring the total of the rounded down sums of x values through them is at least the number of calls crossing them.

Lemma 2.9 (Half-integer rounding of crossing calls) *Given a set of m mutually crossing calls on a ring and a route function x such that $x_{i1} + x_{i2} = 1, \forall$ calls i . There is a half-integer route function x' and a call j such that*

- (i) $x'_{j1} = x'_{j2} = 0$,
- (ii) $x'_{i1} + x'_{i2} = 1, \forall$ calls $i \neq j$
- (iii) *the sum of the x' values at an edge e is at most the sum of the x values rounded down.*

Proof: For the call j , one of whose end vertices is incident on the consecutive edges identified in the proof of Corollary 2.8, set $x'_{j1} = x'_{j2} = 0$. Now, consider any pair of edges in the ring. For every edge, assign a capacity equal to the rounded down sum of the x values through it. By the argument above, for every cut with two edges condition (ii) of Theorem 2.6 is satisfied.

In [13] it is shown that if condition (ii) of Theorem 2.6 holds for cuts that separate the planar graph into two connected components then it also holds for all cuts. For a ring, a cut with 2 edges separates it into two connected components while a cut with more than 2 edges separates it into more than 2 connected components. Therefore, condition (ii) of Theorem 2.6 must hold for all cuts in the ring. Hence, the existence (and construction) of x' follows from it (as the Okamura-Seymour theorem is also constructive).

□

Lemma 2.9 in conjunction with Corollary 2.5 yields an integer rounding that is close to the fractional optimum by an additive constant of 2.

Finally, we can state the performance guarantee of the rounding scheme of crossing calls as follows:

Corollary 2.10 (Rounding class C calls) *Given a set of class C calls $S_{(c)}$ on a ring with a corresponding route function x^* . There exists a $\{0, 1\}$ -route function x' such that*

(i) for every edge e the sum of x' values of routes through it is at most the rounded down value of the sum of the x^ values.*

*(ii) $x'(S_{(c)}) = \sum_{i \in S_{(c)}} (x'_{i1} + x'_{i2}) \geq \sum_{i \in S_{(c)}} (x^*_{i1} + x^*_{i2}) - 2 = x^*(S_{(c)}) - 2$.*

Proof: Lemma 2.9 shows a rounding of x^* to half integer values x losing one on the sum of the x^* values. If all calls got x values in $\{0, 1\}$ we are already done by setting these x values to be x' values. Otherwise, applying Corollary 2.5 on those calls that got x values 0.5 for both their variables we get a $\{0, 1\}$ route function x' . The sum of the x' values is at most one less than the sum of the x values. Thus, in total we lose at most 2 from the sum of x^* values. Condition (i) follows from Lemma 2.9.

This settles the corollary.

□

2.2.3 Assembling the Pieces

In this section, we shall piece together the different parts for solving the RCAC problem. Starting from the optimal fractional solution x^* to the

relaxed LP, we adjust the values such that x^* is in the canonical form with respect to parallel and crossing calls, as set forth in the beginning of Section 2.2. If there is a class D call then make it a class B call by setting the lower of the two indicator variables to zero. Next, perform the rounding on class B and class C calls as described in the Corollaries 2.3 and 2.10. For class B calls the sum of the rounded values at any edge is at most the rounded up value of the sum of the original x^* values and for class C calls it is at most the rounded down value of the sum of the original x^* values. Thus, combining the two sums at an edge will satisfy its capacity constraint. In other words, the rounded solution is a feasible one. As regards the objective value,

$$\begin{aligned}
OPT^* &= x^*(S_{(a)}) + x^*(S_{(b)}) + x^*(S_{(c)}) + x_{D1}^* + x_{D2}^* \\
&< x^*(S_{(a)}) + \lceil x^*(S_{(b)}) + x_{D1}^* \rceil + x^*(S_{(c)}) + 0.5, \quad x_{D2}^* < 0.5 \\
&\leq x'(S_{(a)}) + x'(S_{(b)}) + x'_{D1} + 1 + x'(S_{(c)}) + 2 + 0.5 \\
(x^*(S_{(a)}) = x'(S_{(a)}) = 0, \text{ rounding of class D call with the class B calls}) \\
&= x'(S_{(a)}) + x'(S_{(b)}) + x'_{D1} + x'(S_{(c)}) + 3.5 \\
&= x'(S) + 3.5, \quad S = S_{(a)} \cup S_{(b)} \cup S_{(c)} \cup \{D\}, \quad x'_{D2} = 0
\end{aligned}$$

But, $\lfloor OPT^* \rfloor$ is an upper bound on the objective value of the integer linear program. Therefore, the rounded solution is at most 3 away from an integer optimal solution to the ILP. Yielding,

Theorem 2.11 (“Almost” optimal RCAC solution) *Given an instance of RCAC: a set S of m calls on a ring $C = (V, E)$ with integer edge capacities $w(e)$, $e \in E$. There is a polynomial time algorithm that produces a feasible solution routing at most 3 fewer calls compared to an optimal solution.*

2.2.4 A PTAS for RCAC

Theorem 2.12 *Given $\epsilon > 0$ and an instance of RCAC on a ring there is a polynomial time algorithm that achieves an approximation ratio of $1 - \epsilon$.*

Proof: Let the rounding method output a feasible solution A . We know that $|A| \geq OPT^* - 3.5$. If $OPT \leq OPT^* \leq \frac{3.5}{\epsilon}$, a constant, then we can try all possible $\binom{m}{k}$ subsets of the call set that have cardinality k , $1 \leq k \leq \frac{3.5}{\epsilon}$. For each call in a subset, we can try the two possible routings. Finally, output the solution which has maximum cardinality and is feasible. Plainly,

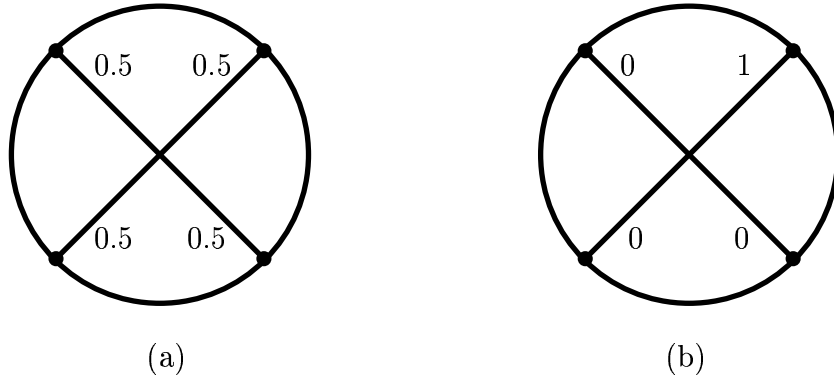


Figure 5: (a) an optimal fractional solution. (b) an optimal integral solution. A value near a chord indicates the route of the call lying on the corresponding side gets that value.

this takes time $O(m^{O(\frac{3.5}{\epsilon})})$. On the other hand, if $OPT^* \geq \frac{3.5}{\epsilon}$,

$$\begin{aligned}
 |A| &\geq OPT^* - 3.5 \\
 &\geq OPT^* - \epsilon \cdot OPT^* \\
 &\geq (1 - \epsilon)OPT^* \\
 &\geq (1 - \epsilon)OPT
 \end{aligned}$$

□

2.2.5 Integrality Gap between ILP and Its Relaxation

It is instructive to study the difference in objective values of the optimal solutions for the ILP and its relaxation for the RCAC problem. For the various instances that we looked at, the objective value of the relaxed LP was at most one away from that for the ILP. It is an open problem to settle the conjecture that for all instances the difference between the objective values of the optimal solutions of the relaxed and integer LPs is at most one. The simplest example for which the conjecture holds is when there are two calls on a ring with edge capacities one. The optimal fractional objective value of 2 is obtained when all the four indicator variables are assigned value 0.5. In contrast, the ILP attains an optimal objective value of 1 by assigning one of the indicator variables 1 and all the others 0. Refer to Figure 5.

3 PCAC on Rings

We turn to the pre-routed variant of the call control problem in rings. In addition to a fixed route for each call which is specified in the input, every call has a non-negative profit associated with it. This problem will be naturally referred to as Pre-routed Call Admission Control Problem, abbreviated PCAC. Formally, the PCAC problem is the following:

Input Instance of PCAC: A ring $R = (V, E)$ on n vertices. A call set S of m calls together with, for each call, one of the two paths as its pre-specified route, a profit function $p : S \rightarrow \mathbb{Z}_0^+$ and a capacity function $w : E \rightarrow \mathbb{Z}^+$.

For the PCAC variant, a *feasible solution* is a subset $S' \subseteq S$ such that the number of routes of calls in S' through an edge $e \in E$ is at most the capacity $w(e)$ of the edge. The profit of the feasible solution S' , denoted $p(S')$, is the sum of the profits of the calls in S' .

Objective and Output of PCAC: A feasible solution $OPT \subseteq S$ with maximum possible profit.

As in the approach to solving RCAC, we shall formulate this problem as an ILP and then show a rounding mechanism. However, unlike for the RCAC variant we obtain only a 2-approximation for the general problem with arbitrary profits. In an earlier work [1], it was shown that if all the profits are equal there is a polynomial time algorithm to solve PCAC that runs in time $O(m^2n \log m)$. The complexity question is still left open for this weighted version. It is satisfying to note that for two important special cases we can prove polynomial time solvability. This leaves one class of PCAC, namely when the profits of pairs of paths do not satisfy the restricted profits property (see Subsection 3.3.3), unresolved in terms of complexity.

3.1 An ILP for PCAC

Let the call set be $S = \{\{u_i, v_i\} : i = 1, 2, \dots, m\}$. Since the routes for each call are specified with the call set we have exactly one indicator variable x_i for call i corresponding to whether the call is accepted and routed along this path. Let $S_e = \{i : \text{call } i \text{ is routed through edge } e\}$, for edge $e \in E$. The

ILP can now be stated as:

$$\begin{aligned} & \max \sum_{i=1}^m p(i) \cdot x_i \\ & \text{subject to} \\ & \sum_{i \in S_e} x_i \leq w(e), \quad e \in E \\ & x_i \in \{0, 1\} \end{aligned}$$

As in RCAC, we shall call the vector x a route function. But here the interpretation would be whether a call i is accepted ($x_i = 1$) or not ($x_i = 0$). For the relaxed LP, the fractional values of a feasible solution will be interpreted as a function that assigns these fractional values to the routes of the calls. A route function is called *feasible* if it satisfies the capacity constraints on the edges.

3.2 A 2-approximation

We shall start with an easy corollary to Lemma 2.1 and then show a weighted version of it which will be essential in deriving the 2-approximation.

Corollary 3.1 *Let S be a set of paths on a line $L = (V, E)$ and $w : E \rightarrow \mathbb{Z}^+$ a capacity function on its edges. Given a route function $x_{max} : S \rightarrow [0, 1]$ such that the sum of the x_{max} values at an edge e is at most $w(e)$ and further $x_{max}(S) \geq x(S)$, for any feasible route function x . Then, \exists a $\{0, 1\}$ -route function $x'_{max} : S \rightarrow \{0, 1\}$ such that $\forall e, \sum_{i \in S_e} x'_{max}(i) \leq w(e)$ and $x'_{max}(S) = x_{max}(S)$.*

Proof: The existence of a feasible x'_{max} such that $x'_{max}(S) \geq \lceil x_{max}(S) \rceil$ follows directly from Lemma 2.1. But by hypothesis, $x_{max}(S)$ is the greatest possible sum among all feasible route functions. Therefore, $x_{max}(S) \geq x'_{max}(S)$ and thus, $x'_{max}(S) = x_{max}(S)$.

□

When the set of paths on the line have profits p associated with them a similar statement is true with respect to the sum of the profits. However, this does not follow from the rounding scheme described in Lemma 2.1 but from the theory of totally unimodular matrices and network flows. A matrix A is said to be totally unimodular if the determinant of every square sub-matrix

of A (including A , if A is a square matrix) is 0, 1, or -1 . A consequence of a matrix A being totally unimodular is that if it appears as the constraint matrix of a linear program $\max\{c^T x : Ax \leq b, 0 \leq x \leq 1\}$, the LP has an integer optimal solution whenever b is integral (see, for example, Corollary 5.20b in [15, p. 76]). It is a well known fact that a $(0, 1)$ -matrix in which the ones appear consecutively in every column (or row) is totally unimodular. From these observations, we have the following weighted version of Corollary 3.1.

Lemma 3.2 *Given a set of paths S with profits $p : S \rightarrow \mathbb{Z}_0^+$ on a line $L = (V, E)$ and a capacity function $w : E \rightarrow \mathbb{Z}^+$. There exists a $\{0, 1\}$ -route function x' such that*

- (i) x' is feasible
- (ii) $\sum_{i \in S} p(i) \cdot x'(i) \geq \sum_{i \in S} p(i) \cdot x(i)$, for every feasible route function x
- and (iii) x' can be computed efficiently.

The 2-approximation for the PCAC problem is rather simple. Identify an edge e on the ring which has the least capacity $w(e) = w_{min}$. Consider the line obtained by removing the edge e from the ring and the set of calls that are not routed through e , namely $S \setminus S_e$. Lemma 3.2 asserts that there is a $\{0, 1\}$ -route function on $S \setminus S_e$ that achieves the optimum profit and is feasible as well. Denote the set of calls in this route function that are assigned unity by $OPT_{\bar{e}}^*$. Arrange the set of calls in S_e in descending order of their profits. Pick the first $w(e)$ calls in this order and call the set OPT_e^* . Both the sets $OPT_e^*, OPT_{\bar{e}}^*$ are feasible solutions to PCAC. Let OPT represent an optimal solution to PCAC on the ring. Trivially, the profit of calls in OPT routed through e , $p(OPT_e) \leq p(OPT_e^*)$. Further, profit of calls in OPT not routed through e , $p(OPT_{\bar{e}}) \leq p(OPT_{\bar{e}}^*)$. Hence,

$$\begin{aligned} p(OPT) &= p(OPT_e) + p(OPT_{\bar{e}}) \\ &\leq p(OPT_e^*) + p(OPT_{\bar{e}}^*) \\ &\leq 2 \max\{p(OPT_e^*), p(OPT_{\bar{e}}^*)\} \end{aligned}$$

By choosing the set which has maximum profit among $\{OPT_e^*, OPT_{\bar{e}}^*\}$ we get a 2-approximation algorithm.

The above algorithm and the analysis can be modified to get a better ratio when the maximum of the lengths of the routes of calls is bounded by

L . Let OPT be the set of calls in an optimal set and $p(OPT)$ their profit. For every edge e in the ring, add up the profits of calls that are routed through it. Let this sum be W . In this sum, each call in OPT has its profit added exactly the length of its route many times. This alternative way of counting W gives us $W = \sum_{i \in OPT} l_i \cdot p(i)$, where l_i is the length of the route of call i . But, $l_i \leq L, \forall i$. Therefore, $W \leq Lp(OPT)$. There are n edges in the ring. Invoking an averaging argument, there must exist an edge at which the sum of the profits of calls which are routed through it is at most $Lp(OPT)/n$. Thus, the profit of calls routed not through e must sum up to at least $(n - L)p(OPT)/n$. But by Lemma 3.2 we can find an optimal profit set of calls that were not routed through e . Thus, the approximation ratio of the algorithm that computes the optimal profit obtained by excluding paths that pass through a particular edge and outputs the highest profit among the iterations for the n edges of the ring, is at most $n/(n - L)$. For values of $L < n/2$, this is better than 2.

3.3 Optimal Algorithms for Special Cases of PCAC

In this subsection, we consider three special cases of PCAC on rings and show optimal algorithms that run in time polynomial in the size of the input. The various special cases are:

- (i) Calls have routes of equal length and their profits are arbitrary.
- (ii) Calls have “proper” routes (defined later) and profits are arbitrary.
- (iii) Calls have arbitrary routes. The profit of a call whose route is contained in that of another is at least as great as the profit of the latter.

(i) is in fact a special case of (ii). We have identified them as separate cases for reasons of presentation. Our algorithms for all these cases are based on network flow techniques and we shall use the following theorem from [3, p. 315] to derive the results.

Theorem 3.3 *Any linear program that contains (a) at most one +1 and at most one -1 in each column or (b) at most one +1 and at most one -1 in each row, can be transformed into a minimum cost flow problem.*

Theorem 3.3 implies that such a linear program can be solved in strongly polynomial time but also the optimal solution vector to the linear program is

integral if the right hand side is an integral vector. Since the edge capacities are integral in our instances, we shall see that we obtain integral optimal solutions for them.

3.3.1 Calls with Paths of Equal Length

For convenience, let us assume that no two calls which have the same end points have been assigned the same route. We shall drop this condition later on. Assume that the routes of all calls have equal length of L . Let the vertices of the ring be numbered $0, 1, \dots, n-1$ in a clockwise fashion and edge i be incident on vertices i and $i+1 \pmod n$, $i = 0, 1, \dots, n-1$. Let the call set be rearranged such that call i is routed by a path containing vertices i through $(i+L) \pmod n$, $i = 0, 1, \dots, n-1$. If no such call appears in the original call set then introduce such a call with profit 0. This does not alter the original instance. With this rearrangement of indices, for any edge j , precisely the following calls pass through it: namely, calls with indices $(j-L+1) \pmod n$ through j . For $j \geq L-1$, this implies all calls with indices $j-L+1$ through j . For $j < L-1$, all calls with indices 0 through j and those with indices $(j-L+1) \pmod n$ through $n-1$. Thus, we can rewrite the relaxation of the ILP in Section 3.1 as:

$$\begin{aligned} & \max \sum_{i=0}^{n-1} p(i) \cdot x_i \\ & \text{subject to} \\ & \sum_{i=j-L+1}^j x_i \leq w(j), \quad n-1 \geq j \geq L-1 \\ & \sum_{i=0}^j x_i + \sum_{i=(j-L+1) \pmod n}^{n-1} x_i \leq w(j), \quad 0 \leq j < L-1 \\ & 0 \leq x_i \leq 1, \quad i = 0, 1, \dots, n-1 \end{aligned}$$

Now, define

$$\begin{aligned} X(-1) &= 0 \\ X(k) &= \sum_{i=0}^k x_i, \quad k = 0, 1, \dots, n-1. \end{aligned}$$

Substituting these new variables in the above LP we obtain (unless we use mod n , -1 is NOT $n-1$):

$$\begin{aligned} & \max \sum_{i=0}^{n-1} p(i) \cdot (X(i) - X(i-1)) \\ & \text{subject to} \\ & X(j) - X(j-L) \leq w(j), \quad n-1 \geq j \geq L-1 \end{aligned}$$

$$\begin{aligned}
&X(j) + X(n-1) - X((j-L) \bmod n) \leq w(j), \quad 0 \leq j < L-1 \\
&0 \leq X(i) - X(i-1) \leq 1, \quad i = 0, 1, \dots, n-1 \text{ corresponding to } 0 \leq x_i \leq 1 \\
&X(-1) = 0
\end{aligned}$$

Naturally, for integer solutions, $X(n-1)$ is an integer between 0 and n . Thus, we can set $X(n-1) = t$, for some integer $t, 0 \leq t \leq n$. This, reduces the constraint matrix to one where each row has at most one +1 and one -1 by taking $X(n-1)$ to the right hand side. Also, since $w(e)$ is an integer, so is $w(e) - t$, for integer t . That the above LP has an integer optimal solution can be deduced from Theorem 3.3 (see comments appearing immediately after the theorem). Moreover, integral optimal solutions can be obtained using network flow techniques. Integer solution for the modified LP implies integer solutions for the original LP as $x_i = X(i) - X(i-1), i = 0, 1, \dots, n-1$. Note also that if X_t^* denotes a feasible vector for the above LP with $X(n-1) = t$ then $\lambda X_{t_1}^* + (1-\lambda)X_{t_2}^*$ is a feasible solution to the LP when $X(n-1) = \lambda t_1 + (1-\lambda)t_2$. Thus, the approach to solve the original problem is:

Step 1: Set $P_1 = \lfloor \frac{n}{2} \rfloor, P_2 = P_1 + 1, P_{min} = 0, P_{max} = n$.

Step 2: If for $X(n-1) = P_2$ the LP is infeasible and for $X(n-1) = P_1$ it is feasible, set $P_{max} = P_1$. Else, if for $X(n-1) = P_1$ the LP is infeasible, set $P_{max} = P_1 - 1$. Otherwise, the LP is feasible for both $X(n-1) = P_1$ and $X(n-1) = P_2$. If $X_{P_1}^*$ has a larger objective value than $X_{P_2}^*$ set $P_{max} = P_1$. If $X_{P_1}^*$ has a smaller objective value than $X_{P_2}^*$ set $P_{min} = P_2$. If the objective values are equal then set $P_{min} = P_1$ and $P_{max} = P_2$.

Step 3: If $P_{max} - P_{min} \leq 1$ output one of the solution vectors $X_{P_{min}}^*$ or $X_{P_{max}}^*$, whichever is feasible and has larger objective value. Else, set $P_1 = \lfloor \frac{P_{min} + P_{max}}{2} \rfloor$ and $P_2 = P_1 + 1$. Go to Step 2.

In the foregoing instance we had assumed that no two calls had the same route if they shared the same end points. This can be easily patched. If the call set had k calls all routed along the same path, say, from vertex i to $(i+L) \bmod n$ then we create k indicator variables $x_i^{(j)}, j = 1, 2, \dots, k$, corresponding to the k copies. The new order of the indices of the calls looks as follows: the distinct routes of the calls are ordered as before and the copies of each of the routes are ordered between themselves arbitrarily. The variable transformation described above can be done according to this new order of the indices of the calls. It is easy to see that the arguments continue to hold for this case too. It should be remarked that the profits of calls play no role in the integrality of the optimal vector. In particular, when two calls

are routed along the same path they could have arbitrary profits.

3.3.2 Calls with “Proper” Routes

A circular arc graph is defined on a set of open arcs on a circle. The vertices correspond to the individual arcs and an edge is drawn between two vertices if the corresponding arcs intersect. A circular arc graph is said to be *proper* if no arc is contained in another. Assume that the input to PCAC is such that no route of a call is strictly contained in that of another. We call such a set of routes proper. The result given in the previous subsection is in reality a special case of proper routes. However, we can deal with this instance also in a similar manner to the last one. The proper-routes-for-calls assumption implies that we can order the calls according to the increasing order of their clock-wise end points of their routes. Once the indices of calls are rearranged according to this order it follows that through an edge e one of the following occurs:

- (a) Calls with indices $1, 2, \dots, k$, $1 \leq k \leq m$ pass through it.
- (b) Calls with indices $1, 2, \dots, k_1, k_2, k_2 + 1, \dots, m$, $1 \leq k_1 < k_2 \leq m$ pass through it.
- (c) Calls with indices $k, k + 1, \dots, m$, $1 \leq k \leq m$ pass through it.
- (d) Calls with indices $k_1, k_1 + 1, \dots, k_2$, $1 < k_1 \leq k_2 < m$ pass through it.

Exactly the same set of transformations as before show that for this case too one can derive an optimal integer solution. The constraints for edges that fall into category (a), (b), (c) and (d) look respectively as $X(k) \leq w(e)$, $X(k_1) + X(m) - X(k_2 - 1) \leq w(e)$, $X(m) - X(k - 1) \leq w(e)$, $X(k_2) - X(k_1 - 1) \leq w(e)$. A binary search, as given in the previous subsection, for values of $X(m)$ between 0 and m outputs an optimal integral solution. Notice that if the set of calls were mutually crossing then the routes are proper. Thus, the special case of mutually crossing calls can also be solved optimally.

3.3.3 Calls with Restricted Profits

In this case, the profits assigned to calls are restricted. For any pair of parallel calls, the profit of the call whose route is strictly contained in that of the other is at least as great as the profit of the latter.

This case can be solved in polynomial time as follows. We start by solving the relaxed ILP, obtaining a fractional optimal solution vector x^* . For any pair of parallel calls i and j , with fractional x^* values and the route of i contained in that of j , we carry out a transformation similar to the one described in Section 2.2 for parallel calls. This transformation ensures that either $x_i^* = 1$ or $x_j^* = 0$. Since by assumption $p(i) \geq p(j)$, the objective value with this transformation is unchanged. At the end of this transformation, we have fractional x^* values only for calls that are mutually crossing. Now, add the constraint that the number of these crossing calls is the rounded down value of the sum of their new x^* values. By the result from the previous subsection, we can find an integer optimal solution that accepts as many calls among them as the rounded down sum of the x^* values and such that the number of these calls that pass through any edge e is at most $w(e)$ minus the number of calls that pass through edge e and whose new x^* values after the transformation were set to 1. The optimal solution to the original problem then consists of those calls that received x^* values 1 in the transformation step and those calls that are in the optimal solution for the crossing calls. It is easy to verify that this is indeed an optimal integral solution. Note that when all calls have equal profits, the problem studied in [1], they form a special case of restricted profits as defined above.

3.4 The Dual LP

It is interesting to note that the dual of the ILP in Section 3.1 is rather special: an integer optimal solution to it can be found in polynomial time. The details follow.

Let the calls be referred to by their indices $1, 2, \dots, m$ and the edges numbered $1, 2, \dots, n$. Define $E_i = \{j : \text{edge } j \text{ is contained in route of call } i\}$. The dual LP can be stated as:

$$\begin{aligned} & \min \sum_{i=1}^m y_i + \sum_{j=1}^n w(j) \cdot z_j \\ & \text{subject to} \\ & y_i + \sum_{j \in E_i} z_j \geq p(i), \quad i = 1, 2, \dots, m \\ & y_i, z_j \geq 0, \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n \end{aligned}$$

We carry out the, by now familiar, variable substitutions. Namely, $Z(j) = \sum_{k=1}^j z_k$, $j = 1, 2, \dots, n$. Also, set $Z(0)=0$. For a call i , let r_i be the highest

indexed edge in its route such that edge $r_i + 1$ is not in the route and l_i the least indexed edge in the route such that edge $l_i - 1$ is not in the route (where, for $r_i = n$, we treat $r_i + 1$ as 1 and for $l_i = 1$, $l_i - 1$ as n). For a call i whose route does not contain both edges 1 and n , we have:

$$\sum_{j \in E_i} z_j = \sum_{j=l_i}^{r_i} z_j = Z(r_i) - Z(l_i - 1)$$

and otherwise:

$$\sum_{j \in E_i} z_j = \sum_{j=1}^{r_i} z_j + \sum_{j=l_i}^n z_j = Z(r_i) + Z(n) - Z(l_i - 1)$$

To take care of the variables y_i , we make the variable substitution:

$$Y(i) = y_i + Z(r_i), \quad i = 1, 2, \dots, m.$$

The dual LP can now be rewritten with each constraint having at most one +1 and one -1:

$$\min \sum_{i=1}^m (Y(i) - Z(r_i)) + \sum_{j=1}^n w(j)(Z(j) - Z(j-1))$$

subject to

$$Y(i) - Z(l_i - 1) \geq p(i), \text{ if call } i \text{ is not routed through edges 1 and } n$$

$$Y(i) - Z(l_i - 1) \geq p(i) - Z(n), \text{ if call } i \text{ is routed through edges 1 and } n$$

$$Y(i) - Z(r_i) \geq 0, \quad i = 1, 2, \dots, m, \text{ corresponding to non-negativity of } y_i$$

$$Z(0) = 0, Z(j) - Z(j-1) \geq 0, \quad j = 1, 2, \dots, n, \text{ corresponding to non-negativity of } z_j$$

A feasible solution to the dual is obtained by letting $Y(i) = p(i)$, $i = 1, 2, \dots, m$ and $Z(j) = 0$, $j = 1, 2, \dots, n$. Thus, the objective value of the dual LP is at most $P = \sum_{i=1}^m p(i)$. But, in an optimal solution,

$$\begin{aligned} Z(n) &= \sum_{i=1}^n (Z(i) - Z(i-1)) \\ &\leq \sum_{i=1}^n w(i)(Z(i) - Z(i-1)), \quad w(i) \geq 1 \\ &\leq \sum_{i=1}^m (Y(i) - Z(r_i)) + \sum_{j=1}^n w(j)(Z(j) - Z(j-1)) \\ &\leq P \end{aligned}$$

Thus, to obtain an integral optimal solution for the dual we perform a binary search on $Z(n) = p$, for integer values of p between 0 and P . This, reduces the dual to the form as required in Theorem 3.3. Whence, we can solve it by network flow techniques. The fastest known minimum cost flow algorithm runs in time $O(m \log n(m + n \log n))$ [14], where n and m are the number of vertices and edges in the network, respectively. For the binary

search we have an extra factor of $\log P \leq \log(mP_{max})$, where P_{max} is the maximum of the profits of calls. An integral optimal solution to the dual can be computed in $O(m^2 \log^2 m \log(mP_{max}))$ time.

3.5 Algorithms for Special Cases

The special cases considered in Subsection 3.3 restricted the *nature* of calls, like equal length routes or proper routes etc. In this subsection, by contrast, we shall place restrictions on the different *numerical parameters* that appear in the input, namely the edge capacities, profits and lengths. More concretely, we study instances where either:

- (a) The minimum capacity on edges is bounded by a constant, or
- (b) The profits are proportional to the length of the route of calls, or
- (c) The length of route of calls is bounded by a constant.

For the first, a polynomial time optimal algorithm is shown. For the second, a polynomial time approximation scheme (PTAS) is derived while for the last, an optimal algorithm and a PTAS (when the edge capacities are sufficiently high) are given. The PTAS for the last case is useful to obtain a fast approximation to the optimal solution.

Minimum edge capacity bounded by a constant: Let the minimum edge capacity in the ring be W_{min} , a constant. Let e_{min} be an edge with capacity W_{min} . Note that any optimal solution to the PCAC problem routes at most W_{min} calls through the edge e_{min} . Let the number of calls with routes through e_{min} be $m_{e_{min}} \leq m$, where m is the total number of calls. It is straightforward to see that the number of subsets of calls routed through e each with cardinality at most W_{min} is $O((m_{e_{min}} + 1)^{W_{min}}) = O((m + 1)^{W_{min}}) = O(\text{poly}(m))$. For each such subset, by Lemma 3.2, we can find in polynomial time a maximum profit feasible call set among the rest of the calls not routed through e_{min} and that satisfies the edge capacity constraints. The optimal set for the instance is the one with maximum profit among this enumeration. The running time of the algorithm is $O(\text{poly}(m, n))$.

Calls with profits proportional to length of routes: Here we look at a “natural” profit assignment to a call; viz., the profit is the length of the route of the call (could as well be proportional to it). Further, we consider

only routes of length at most $\lfloor n/2 \rfloor$, where n is the number of edges in the ring.

Identify two edges e and e' that have $\lfloor n/2 \rfloor$ vertices between them. Since every route is of length at most $\lfloor n/2 \rfloor$, none of them pass through both e and e' . Now, compute an optimal fractional solution to the LP. Partition the set of calls S as $S_{\bar{e}}$ and $S_e = S \setminus S_{\bar{e}}$, where $S_{\bar{e}}$ is the set of calls whose routes do not pass through e . Since the set of routes in $S_e(S_{\bar{e}})$ do not pass through edge $e'(e)$, by Lemma 3.2, there is a $\{0, 1\}$ -routing for each of them separately such that the sum of the rounded values is at most the rounded up value of their x^* sums. The rounding maintains at least the sums of the profit. However, it could violate the feasibility at an edge by 1.

A *minimum profit circle cover* is a set of routes such that at least one route passes through every edge and the sum of the profits of the routes is minimum possible. It is easy to see that in a minimum profit circle cover no route is strictly contained in another. Thus, in a minimum profit circle cover no edge has more than 2 routes passing through it. For otherwise, if 3 routes that are not contained in each other pass through an edge then by removing the route which is contained in the union of the other two routes we have a smaller profit circle cover. Computing a minimum profit circle cover for the routes of calls whose x^* values were rounded to 1, can be done in linear time [4].

A feasible solution is obtained by discarding the routes in a minimum profit circle cover from the infeasible rounded solution. Since every edge in a minimum profit circle cover has at most 2 routes passing through it, the profit of the minimum profit circle cover is at most $2n$.

Given an $\epsilon > 0$, fix the constant $K = 2\lceil 1/\epsilon \rceil$. If the objective value of the fractional optimal solution $p(OPT^*) \leq Kn$, then there exists an edge in the ring through which at most K paths pass in any optimal integral solution. It follows from this argument; the profit of a feasible solution can be counted in two ways: (i) sum of the profits of calls in the solution and (ii) total of the number of routes through every edge. If $p(OPT^*) \leq Kn$ then $p(OPT) \leq Kn$, where OPT is an optimal solution to PCAC. $p(OPT)$ is the sum of the number of routes through every edge in the optimal solution. There are n edges, thus the number of routes through one of the edges should

be at most K . Thus, we can find an optimal solution to the instance by assigning for each edge in turn an edge capacity equal to K . By the previous special case, an optimal solution can be found in polynomial time when the minimum edge capacity is bounded by a constant.

On the other hand, let $p(OPT^*) > Kn$. We have $p(OPT^*) > 2\lceil 1/\epsilon \rceil n$. Therefore, the feasible solution computed by throwing out a minimum profit circle cover from a rounded solution has a profit at least $p(OPT^*) - 2n \geq p(OPT^*) - \epsilon p(OPT^*)$. Or the profit of the feasible solution is at least $(1-\epsilon)p(OPT^*)$. But $p(OPT^*)$ is an upper bound on the profit of any feasible solution. Thus, we have a PTAS for this special instance.

Calls with bounded route lengths: Let the maximum length of a route of a call be L , a constant. Since we have shown a polynomial time optimal algorithm for bounded minimum edge capacity, we shall assume that the minimum edge capacity is greater than a given constant.

There is a dynamic programming approach that solves this instance optimally. However, the running time is exponential in L . The dynamic program works as follows: Fix an edge e in the ring. Put the route of a call which contains e , has length i and contains exactly r consecutive edges that lie clockwise to e , into the set $S_{i,r}$, $i = 1, 2, \dots, L$, $r = 0, 1, 2, \dots, i - 1$. There are $O(L^2)$ such sets. All these sets have a cardinality of at most m , the total number of calls. The dynamic program is to try all possible at most m values for each set, $S_{i,r}$. In each set, we naturally pick the calls in decreasing order of their profits. For the rest of the paths that do not pass through e we can apply the result of Lemma 3.2 with the edge capacities reduced by the appropriate number of paths. Every entry of the DP table is the profit that is achieved for a particular way of fixing values for $S_{i,r}$. Finally, we pick that entry of the DP table that has the maximum profit. The running time is $O(m^{L^2})$.

If the edge capacities are sufficiently high then we can get a good, fast approximation to an integral optimal solution from the fractional optimal solution. The details are presented below.

For a call i and an edge e through which its route passes, let $S_{i,e}$ denote the set of calls with indices less than i and whose routes contain e . Let

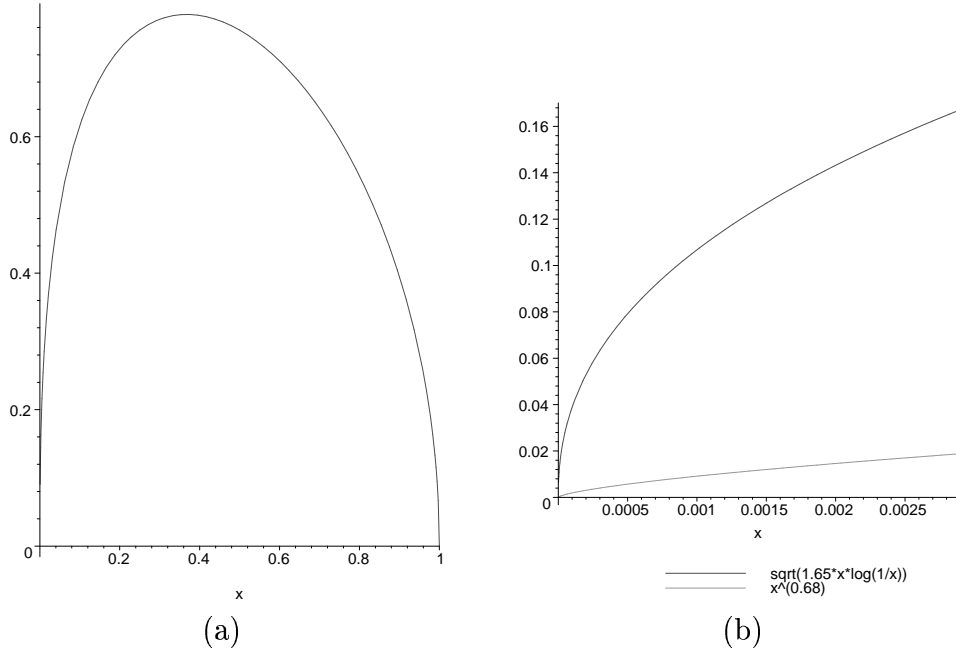


Figure 6: (a) Plot of $f(x) = \sqrt{1.65x \ln(1/x)}$, $x \in [0, 1]$. (b) Comparison of the functions $\sqrt{1.65x \ln(1/x)}$ and $x^{0.68}$, $x \in [0, 0.00288]$.

$x^* = (x_1^*, x_2^*, \dots, x_m^*)$ be an optimal fractional solution to the relaxed LP of Section 3.1. Let OPT and OPT^* denote the objective value of an optimal integral and fractional solution respectively. An adaptation of a randomized rounding method described in [6] gives a polynomial time randomized approximation scheme.

Define $f(\delta) = \sqrt{1.65\delta \ln(1/\delta)}$. Given $1/3 \geq \epsilon' > 0$, let δ' be such that: $\epsilon' = f(\delta')$ and $\delta' \leq 1/e$. We shall need the following numerical details for the analysis, $f(0.00288) = 0.1667$. Also, for $0 < \delta \leq 0.00288$, $f(\delta) > 57\delta$ and $f(\delta) > \delta^{0.68}$. See Figure 6. First, we want to show that $\min\{f(\frac{\delta'}{2L^2}), f((\frac{1}{2L^2})^2)\} \leq (1/L)f(\delta')$.

Case A: $2L^2 \leq \frac{1}{\delta'}$

We have $\ln 2L^2 \leq \ln \frac{1}{\delta'}$. Therefore,

$$\begin{aligned} f\left(\frac{\delta'}{2L^2}\right) &= \sqrt{1.65\left(\frac{\delta'}{2L^2}\right) \ln(2L^2/\delta')} \\ &= (1/L) \sqrt{1.65(\delta'/2)(\ln(2L^2) + \ln(1/\delta'))} \end{aligned}$$

$$\begin{aligned} &\leq (1/L)\sqrt{1.65(\delta'/2)(2 \cdot \ln(1/\delta'))} \\ &= (1/L)f(\delta') \end{aligned}$$

Case B: $2L^2 \geq \frac{1}{\delta'}$ or $1/2L^2 \leq \delta'$

Since $\delta' < 1/e$ and $f(\cdot)$ is increasing in $(0, 1/e)$, $f(1/2L^2) \leq f(\delta')$.

Now,

$$\begin{aligned} f((1/2L^2)(1/2L^2)) &= \sqrt{1.65(1/2L^2)^2 \ln((2L^2)^2)} \\ &= (1/L)\sqrt{1.65(1/4L^2)(2 \ln(2L^2))} \\ &= (1/L)f(1/2L^2) \\ &\leq (1/L)f(\delta') \end{aligned}$$

Let $\delta'' = \min\{\delta'/2L^2, 1/4L^4\}$. Since L and δ' are constants so is δ'' . If the minimum edge capacity is bounded by the constant $1/\delta''$ then we have a polynomial time optimal algorithm. Hence, we may assume the minimum edge capacity w_{min} is greater than $1/\delta''$.

$$\begin{aligned} w_{min}\delta'' &> 1 \\ \text{implying, } w(e)\delta'' &> 1, \quad \forall e \end{aligned}$$

Let $\epsilon = \frac{\epsilon'}{L}$. Let δ be such that $\epsilon = f(\delta)$ and $\delta \leq \delta'$. Clearly, $\delta \geq \delta''$, since $f(\delta) = \epsilon = \epsilon'/L \geq f(\delta'')$. Therefore, $w_{min}\delta \geq w_{min}\delta'' > 1$. Now, we do the following randomized rounding. Set variable x_i corresponding to call i to 1 with probability $(1 - \epsilon)x_i^*$. Call these random variables $Y_i, i = 1, 2, \dots, m$. Consider the following feasible solution to the PCAC problem:

$$Z_i = \begin{cases} 1 & \text{if } Y_i = 1 \text{ and } \sum_{j \in S_{i,e}} Z_j \leq w(e) - 1, \forall e \in E_i \\ 0 & \text{otherwise} \end{cases}$$

We state without proof the following lemma in [6]:

Lemma 3.4 *Let X_1, X_2, \dots, X_m be independent random variables and let $0 \leq \beta_1, \beta_2, \dots, \beta_m \leq 1$ be reals, where for $i \in \{1, 2, \dots, m\}$, $X_i = \beta_i$ with probability p_i , and $X_i = 0$ otherwise. Let $X = \sum_i X_i$ and $\mu = E[X]$. Then*

(i) $\sigma(X) \leq \sqrt{\mu}$.

(ii) For any λ with $0 < \lambda < \sqrt{\mu}$, $Pr[X > \mu + \lambda\sqrt{\mu}] < \exp(-\frac{\lambda^2}{2}(1 - \lambda/\sqrt{\mu}))$.

Now, we prove the following lemma:

Lemma 3.5 *Given $1/3 > \epsilon' > 0$ and $L \geq 2$, let $\epsilon = \epsilon'/L (< 1/6)$ and δ be such that $\epsilon = f(\delta)$ and $\delta < 0.00288$. Then, $Pr[Z_i = 1] \geq (1 - 1.5\epsilon')x_i^*$.*

Proof: Consider an edge $e \in E_i$ of a call i . Let $\pi_{i,e} = \{Z_i = 0 \text{ because } e \text{ was saturated by routes of calls in } S_{i,e} | Y_i = 1\}$. That is, $\pi_{i,e}$ denotes the event that call i was not accepted because edge e has already $w(e)$ calls in $S_{i,e}$ routed through it.

$$\Pr[\pi_{i,e}] = \Pr[\sum_{j \in S_{i,e}} Z_j = w(e) > w(e) - 1]$$

But $Z_j \leq Y_j$, for all j . Therefore,

$$\Pr[\pi_{i,e}] \leq \Pr[\sum_{j \in S_{i,e}} Y_j > w(e) - 1]$$

Equivalently,

$$\Pr[\pi_{i,e}] \leq \Pr[\sum_{j \in S_{i,e}} \frac{Y_j}{\delta w(e)} > \frac{w(e)-1}{\delta w(e)}] \leq \Pr[\sum_{j \in S_{i,e}} \frac{Y_j}{\delta w(e)} > \frac{1-\delta}{\delta}]$$

The random variables $\{Y_j/\delta w(e)\}_{j \in S_{i,e}}$, satisfy the conditions of Lemma 3.4. Let $Y = \sum_{j \in S_{i,e}} \frac{Y_j}{\delta w(e)}$ and $\mu = E[Y]$. By linearity of expectation,

$$\mu = \sum_{j \in S_{i,e}} \frac{1}{\delta w(e)} \cdot (1 - \epsilon)x_j^* = \frac{1-\epsilon}{\delta} \sum_{j \in S_{i,e}} \frac{x_j^*}{w(e)} \leq \frac{1-\epsilon}{\delta}$$

The last inequality follows from the capacity constraint on edge e . Case I: $\mu < (7/8)(1 - \delta)/\delta$. Since $\frac{\sigma(Y)}{\sqrt{\mu}} \leq 1$ by Lemma 3.4,

$$\Pr[\pi_{i,e}] \leq \Pr[Y > \frac{1-\delta}{\delta}] \leq \Pr[|Y - \mu| > \frac{1}{8} \cdot \frac{1-\delta}{\delta} \cdot \frac{\sigma(Y)}{\sqrt{\mu}}] \leq \frac{64\delta^2 \mu}{(1-\delta)^2}$$

The last inequality follows from Chebyshev's inequality. Given that $\delta < 0.00288$, we obtain, $\Pr[\pi_{i,e}] < 57\delta < \epsilon$.

Case II: $\mu \geq (7/8)(1 - \delta)/\delta$. Set λ such that $\mu + \lambda\sqrt{\mu} = (1 - \delta)/\delta$. We now have,

$$\lambda = \frac{\frac{1-\delta}{\delta} - \mu}{\sqrt{\mu}} \geq \frac{\frac{1-\delta}{\delta} - \frac{1-\epsilon}{\delta}}{\sqrt{\frac{1-\epsilon}{\delta}}} = \frac{\epsilon - \delta}{\sqrt{\delta(1-\epsilon)}} \geq \frac{\epsilon - \delta}{\sqrt{\delta}} \geq \frac{56}{57} \frac{\epsilon}{\sqrt{\delta}}$$

Further, $1 - \lambda/\sqrt{\mu} = 2 - (1 - \delta)/(\delta\mu) \geq 6/7$. By Lemma 3.4 on random variables $\{Y_j/\delta w(e)\}_{j \in S_{i,e}}$, we get,

$$\begin{aligned} \Pr[\pi_{i,e}] &\leq \Pr[Y > \mu + \lambda\sqrt{\mu}] < \exp\left(-\frac{\lambda^2}{2}(1 - \lambda/\sqrt{\mu})\right) < \exp\left(-\frac{1}{2}\left(\frac{56}{57}\right)^2 \frac{\epsilon^2}{\delta} \frac{6}{7}\right) \\ &< \delta^{0.68} < \epsilon \end{aligned}$$

Therefore, $\Pr[\pi_{i,e}] < \epsilon$. Now,

$$\begin{aligned} \Pr[Z_i = 0 | Y_i = 1] &= \Pr[\pi_{i,e} \text{ for at least one } e \in E_i] \\ &\leq \sum_{e \in E_i} \Pr[\pi_{i,e}] \\ &< l_i \epsilon, \quad l_i \text{ is length of route of call } i \\ &\leq L \epsilon, \quad \text{since } l_i \leq L \end{aligned}$$

Finally, we obtain,

$$\begin{aligned} \Pr[Z_i = 1] &= (1 - \Pr[Z_i = 0 | Y_i = 1]) \cdot \Pr[Y_i = 1] \\ &\geq (1 - L\epsilon)(1 - \epsilon)x_i^* \\ &= (1 - \epsilon')(1 - \epsilon'/L)x_i^* \\ &\geq (1 - (1 + 1/L)\epsilon')x_i^* \\ &\geq (1 - 1.5\epsilon')x_i^*, \quad L \geq 2 \end{aligned}$$

□

Theorem 3.6 *Given $\epsilon' > 0$ and a PCAC instance on rings with the length of routes of calls bounded by a constant there is a randomized rounding achieving an approximation ratio of $(1 - 3\epsilon')$ with probability at least $1/2$.*

Proof: If the minimum capacity of the edges is bounded by a constant there is a polynomial time optimal algorithm to solve PCAC. Assume the minimum capacity is not bounded by a constant. Then the previous analysis goes through. Let P denote the profit. The expected profit of the randomized rounding is $E[P] = \sum_{i=1}^m p(i) \cdot \Pr[Z_i = 1]$. This is at least $(1 - 1.5\epsilon')OPT^* \geq (1 - 1.5\epsilon')OPT$. But P never exceeds OPT . Therefore, we have $\Pr[P \geq (1 - 3\epsilon')OPT] \geq 1/2$. Hence, the randomized algorithm achieves the desired ratio with a probability of at least $1/2$.

□

Acknowledgements

This work was supported by the Berlin-Zürich Joint Graduate Program “Combinatorics, Geometry, and Computation” (CGC), financed by ETH Zürich and the German Science Foundation (DFG).

References

- [1] U. Adamy, C. Ambuehl, R.S. Anand, and T. Erlebach. Call control in rings. In *Proceedings of the 29th International Colloquium on Automata, Languages and Programming ICALP 2002*, LNCS 2380, pages 788–799, 2002.
- [2] R. Adler and Y. Azar. Beating the logarithmic lower bound: randomized preemptive disjoint paths and call control algorithms. In *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms SODA 1999*, pages 1–10, 1999.
- [3] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network flows: Theory, algorithms and application*. Prentice-Hall, New York, USA, 1993.
- [4] M.J. Atallah, D.Z. Chen, and D.T. Lee. An optimal algorithm for shortest paths on weighted interval and circular-arc graphs, with applications. *Algorithmica*, 14(5):15–26, 1995.
- [5] A. Blum, A. Kalai, and J. Kleinberg. Admission control to minimize rejections. In *Proceedings of the 7th Workshop on Algorithms and Data Structures WADS 2001*, LNCS 2125, pages 155–164, 2001.
- [6] G. Calinescu, A. Chakrabarti, H. Karloff, and Y. Rabani. Improved approximation algorithms for resource allocation. In *Proceedings of the 9th International Conference on Integer Programming and Combinatorial Optimization IPCO 2002*, LNCS 2337, pages 401–414, 2002.
- [7] M.C. Carlisle and E.L. Lloyd. On the k -coloring of intervals. *Discrete Applied Mathematics*, 59:225–235, 1995.
- [8] T. Erlebach. Approximation algorithms and complexity results for path problems in trees of rings. In *Proceedings of the 26th International Symposium on Mathematical Foundations of Computer Science MFCS 2001*, LNCS 2136, pages 351–362, 2001.
- [9] T. Erlebach and K. Jansen. The maximum edge-disjoint paths problem in bidirected trees. *SIAM Journal on Discrete Mathematics*, 14(3):326–355, 2001.

- [10] J.A. Garay, I.S. Gopal, S. Kutten, Y. Mansour, and M. Yung. Efficient on-line call control algorithm. *Journal of Algorithms*, 23:180–194, 1997.
- [11] N. Garg, V.V. Vazirani, and M. Yannakakis. Primal-dual approximation algorithms for integral flow and multicut in trees. *Algorithmica*, 18(1):3–20, 1997.
- [12] C. Nomikos, A. Pagourtzis, and S. Zachos. Minimizing request blocking in all-optical rings. In *IEEE INFOCOM*, 2003.
- [13] H. Okamura and P. Seymour. Multicommodity flows in planar graphs. *Journal of Combinatorial Theory, Series B*, 31:75–81, 1981.
- [14] J.B. Orlin. A faster strongly polynomial minimum cost flow algorithm. *Operations Research*, 41:338–350, 1993.
- [15] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer, Heidelberg, Germany, 2003.
- [16] P.J. Wan and L. Liu. Maximal throughput in wavelength-routed optical networks. In *Multichannel Optical Networks: Theory and Practice*, volume 46 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 15–26. AMS, 1998.