SIAM J. COMPUT.
Vol. 30, No. 6, pp. 2051–2068

A CONSTANT-FACTOR APPROXIMATION ALGORITHM FOR PACKET ROUTING AND BALANCING LOCAL VS. GLOBAL CRITERIA*

ARAVIND SRINIVASAN† AND CHUNG-PIAW TEO‡

Abstract. We present the first constant-factor approximation algorithm for a fundamental problem: the *store-and-forward packet routing problem on arbitrary networks*. Furthermore, the queue sizes required at the edges are bounded by an absolute constant. Thus, this algorithm balances a *global* criterion (routing time) with a *local* criterion (maximum queue size) and shows how to get simultaneous good bounds for both. For this particular problem, approximating the routing time well, even without considering the queue sizes, was open. We then consider a class of such local vs. global problems in the context of covering integer programs and show how to improve the local criterion by a logarithmic factor by losing a constant factor in the global criterion.

Key words. packet routing, approximation algorithms, linear programming, multicommodity flow, rounding theorems, randomized algorithms, covering integer programs, randomized rounding, discrete ham-sandwich theorems

AMS subject classifications. 68Q15, 05C65

PII. S0097539798335596

1. Introduction.

Recent research on approximation algorithms has focused a fair amount on *bicriteria* (or even multicriteria) minimization problems, attempting to simultaneously keep the values of two or more parameters "low" (see, e.g., [11, 21, 22, 29, 30, 32]). One motivation for this is that real-world problems often require such balancing. In this work, we consider a family of bicriteria problems that involve balancing a *local capacity* constraint (e.g., the maximum queue size at the links of a packet routing network, the maximum number of facilities per site in facility location) with a global criterion (resp., routing time, total cost of constructing the facilities). Since these global criteria are NP-hard to minimize even with no constraint on the local criterion, we shall seek good approximation algorithms.

1.1. Packet routing.

Our main result is a constant-factor approximation algorithm for store-and-forward packet routing, a fundamental routing problem in interconnection networks (see Leighton's book and survey [14, 15]); furthermore, the queue sizes will all be bounded by a constant. This packet routing problem has received considerable attention for more than 15 years and is as follows.

DEFINITION 1.1 (store-and-forward packet routing). *We are given an arbitrary N-node routing network (directed or undirected graph) G and a set $\{1, 2, \ldots, K\}$ of packets which are initially resident (respectively) at the (multi-)set of nodes* $\{s_k : 1 \leq$

*Received by the editors March 18, 1998; accepted for publication (in revised form) September 7, 2000; published electronically April 18, 2001. A preliminary version of this work appeared in *Proceedings of the ACM Symposium on Theory of Computing*, El Paso, TX, 1997, pp. 636–643.
http://www.siam.org/journals/sicomp/30-6/33559.html

†Bell Laboratories, Lucent Technologies, 600–700 Mountain Ave., Murray Hill, NJ 07974-0636 (srin@research.bell-labs.com). Part of this work was done while the author was at the School of Computing, National University of Singapore, Singapore 119260 and was supported in part by National University of Singapore Academic Research Fund grants RP950662, RP960620, and RP970607.

‡Department of Decision Sciences, National University of Singapore, Singapore 119260, Republic of Singapore (fbateocp@nus.edu.sg). This author was supported in part by National University of Singapore Academic Research Fund grant RP3970021 and a Fellowship from the Singapore-MIT Alliance Program in High-Performance Computation for Engineered Systems.

$k \leq K\}$ of G. Each packet k is a message that needs to be routed to some given destination node t_k in G. We have to route each packet k from s_k to t_k, subject to the following: (i) *each packet k must follow some path in G;* (ii) *each edge traversal takes one unit of time;* (iii) *no two packets can traverse the same edge at the same unit of time; and* (iv) *packets are only allowed to queue along the edges of G during the routing stage. There are no other constraints on the paths taken by the packets: these can be arbitrary paths in G. The NP-hard objective is to select a path for each packet and to coordinate the routing so that the elapsed time by which all packets have reached their destinations is minimized; i.e., we wish to keep this routing time as small as possible.*

Extensive research has been conducted on this problem: see [14, 15] and the references therein. The most desirable type of algorithm here would, in addition to keeping the routing time and queue sizes low, also be *distributed*: given a set of incoming packets and their (source, destination) values, any switch (node of G) decides what to do with them next, without any other knowledge of the (multi-)set $\{(s_k, t_k) : 1 \leq k \leq K\}$. This would be ideal for parallel computing. (Distributed algorithms in this context are also termed *on-line* algorithms in the literature.) Several such ingenious results are known for specific networks such as the mesh, butterfly, or hypercube. For instance, given any routing problem with N packets on an N-node butterfly, there is a randomized on-line routing algorithm that, with high probability, routes the packets in $O(\log N)$ time using $O(1)$-sized queues [28]. (We let e denote the base of the natural logarithm, and, for $x > 0$, $\lg x$, $\ln x$, and $\ln^+ x$, respectively, denote $\log_2 x$, $\log_e x$, and $\max\{\log_e x, 1\}$. Also, Z_+ will denote the set of nonnegative integers.)

Good on-line algorithms here, however, are not always feasible or required for the following reasons:

- A large body of research in routing is concerned with *fault-tolerance*: the possibility of G being a reasonable routing network when its nodes are subject to (e.g., random or worst-case) faults. See, e.g., Kaklamanis et al. [12], Leighton, Maggs, and Sitaraman [18], and Cole, Maggs, and Sitaraman [6]. In this case, we do not expect good on-line algorithms, since the fault-free subgraph $\hat{G}$ of G has an unpredictable structure. Indeed, a fair amount of research in this area, e.g., [6, 18], focuses on showing that $\hat{G}$ is still a reasonably good routing network under certain fault models, assuming global information about $\{(s_k, t_k)\}$ and the fault structure.
- Ingenious on-line algorithms for specific networks such as the butterfly in the fault-free case [28] are only *existentially* (near-)optimal. For instance, the $O(\lg N)$ routing time of [28] is existentially optimal to within a constant factor, since there are families of routing instances that require $\Theta(\lg N)$ time. However, the worst-case approximation ratio can be $\Theta(\lg N)$. It seems very hard (potentially impossible) to devise on-line algorithms that are near-optimal on *each instance*.
- The routing problem can be considered as a variant of unit-demand multicommodity flow, where all arc capacities are the same, queuing is allowed, and where delivery time is also a crucial criterion. (Algorithms for this problem that require just $O(1)$ queue sizes, such as ours, will also scale with network size.) For such flow problems, the routing problems often have to be run repeatedly. It is therefore reasonable to study *off-line* approximation algorithms, i.e., efficient algorithms that use the knowledge of the network and of

$\{(s_k, t_k)\}$ and have a good approximation ratio.

Furthermore, it seems like a difficult problem to construct on-line routing algorithms for *arbitrary* networks, even with, say, a polylogarithmic approximation guarantee. See Ostrovsky and Rabani [26] for good on-line packet scheduling algorithms, *given the path to be traversed for each packet*.

By combining some new ideas with certain powerful results of Leighton, Maggs, and Rao [16], Leighton, Maggs, and Richa [17], Karp et al. [13], and Lin and Vitter [20], we present the first polynomial-time off-line *constant-factor* approximation algorithm for the store-and-forward packet routing problem. Furthermore, the queue sizes of the edges are bounded by $O(1)$. No approximation algorithms with a sublogarithmic approximation guarantee were known for this problem, to the best of our knowledge. For instance, a result from the seminal work of Leighton and Rao [19] leads to routing algorithms that are *existentially* good. Their network embedding of G ensures that there is some routing instance on G for which their routing time is to within an $O(\lg N)$ factor of optimal, but no good worst-case performance guarantee is known. We may attempt randomized rounding on some suitable linear programming (LP) relaxation of the problem; however, apart from difficulties like controlling path lengths, it seems hard to get a constant-factor approximation using this approach for families of instances where the LP optimal value grows as $o(\lg(N+K))$. Our approach uses the rounding theorem of [13] to select the set of paths that will be used in the routing algorithm of [17]. The analysis involves an interesting trade-off between the "dilation" criterion (maximum path length) and the "congestion" criterion (maximum number of paths using any edge).

1.2. Covering integer programs. Let v^T denote the transpose of a (column) vector v. In the second part of the paper, we continue to address the problem of simultaneously obtaining good bounds on two criteria of a problem. We focus on the NP-hard family of *covering integer programs* (CIPs), which includes the well-known set cover problem. This class of problems exhibits features similar to our packet routing problem: the latter can be formulated as a covering problem with side packing constraints. In CIPs, the packing constraints are upper bound constraints on the variables.

DEFINITION 1.2 (CIPs). *Given $A \in [0,1]^{m\times n}$, $b \in [0,\infty)^m$, and $c \in [0,1]^n$, a CIP seeks to minimize $c^T \cdot x$ subject to $Ax \geq b$, $x \in Z_+^n$, and $0 \leq x_j \leq d_j$ for each j (the $d_j \in Z_+$ are given integers). If $A \in \{0,1\}^{m\times n}$, then we assume without loss of generality (w.l.o.g.) that each b_i is a positive integer. Define $B = \min_i b_i$; w.l.o.g., we may assume $B \geq 1$. A CIP is* uncapacitated *if for all j, $d_j = \infty$.*

It is well known that the two assumptions above are w.l.o.g. (i) If $A \in \{0,1\}^{m\times n}$, then we can clearly replace each b_i by $\lceil b_i \rceil$. (ii) Given a CIP with some $A_{i,j} > b_i$, we can *normalize* it by first setting $A_{i,j} := b_i$ for each such (i,j) and then scaling A and b uniformly so that for all k, ($b_k \geq 1$ and $\max_\ell A_{k,\ell} \leq 1$). This is easily seen to result in an equivalent CIP.

To motivate the model, we consider a concrete CIP example: a facility location problem that generalizes the set cover problem. Here, given a digraph G, we want to place facilities on the nodes suitably so that every node has at least B facilities in its out-neighborhood. Given a cost-per-facility c_j of placing facilities at node j, we desire to place the facilities in a way that will minimize the total cost. It is easy to see that this NP-hard problem is a CIP, with the matrix A having only zeroes and ones. This problem illustrates one main reason for the constraints $\{x_j \leq d_j\}$: for reasons of capacity, security, or fault-tolerance (not many facilities will be damaged if, for instance, there is an accident/failure at a node), we may wish to upper bound the number of facilities that can be placed at individual sites. The more general problem of "file sharing" in a network has been studied by Naor and Roth [24], where again, the maximum load (number of facilities) per node is balanced with the global criterion of total construction cost. For similar reasons, CIPs typically include the constraints $\{x_j \leq d_j : 1 \leq j \leq n\}$. In fact, the case where $d_j = 1$ for all j is quite common.

Dobson [7] and Fisher and Wolsey [8] study a natural greedy algorithm $\mathcal{GA}$ for CIPs. For a given CIP, let OPT denote the value of its optimal *integral* solution. We define $\gamma_1 \doteq \min_{i,j}\{A_{i,j}/c_j : A_{i,j} \neq 0\}$ and $\gamma_2 \doteq \max_j(\sum_{i=1}^m A_{i,j}/c_j)$. Then, it is shown in [8] that $\mathcal{GA}$ produces a solution of value at most $OPT(1 + \ln(\gamma_2/\gamma_1))$. If each row of the linear system $Ax \geq b$ is scaled so that the minimum nonzero entry in the row is at least 1, it is shown in [7] that $\mathcal{GA}$'s output is at most $OPT(1 + \ln(\max_j \sum_{i=1}^n A_{i,j}))$.

Another well-known approach to CIPs is to start with their LP relaxation, wherein each x_j is allowed to be a *real* in the range $[0, d_j]$. Throughout, we shall let y^* denote the LP optimum of a given CIP. Clearly, y^* is a lower bound on OPT. Bertsimas and Vohra [5] conduct a detailed study of approximating CIPs and present an approximation algorithm which finds a feasible solution whose value is $O(y^* \lg m)$ [5]. Previous work of this paper's first author [31] presents an algorithm that computes an $x \in Z_+^n$ such that $Ax \geq b$ and

$$c^T \cdot x \leq a_0 y^* \cdot \max\{\ln^+(mB/y^*)/B, \sqrt{\ln^+(mB/y^*)/B}\} \tag{1.1}$$

for some absolute constant $a_0 > 0$.[1] The bound "$x_j \leq d_j$" may not hold for all j, but we will have for all j that

$$x_j \leq d_j \cdot \left(1 + a_1 \max\left\{\ln^+(mB/y^*)/B, \sqrt{\ln^+(mB/y^*)/B}\right\}\right) \tag{1.2}$$

for a certain absolute constant $a_1 > 0$. A related result is presented in [24] for file-sharing.

If B is "large" (greater than a certain threshold), then these results significantly improve previous results in the "global" criterion of keeping $c^T \cdot x$ small while compromising somewhat on the "local" capacity constraints $\{x_j \leq d_j\}$. This is a common approach in bicriteria approximation: losing a small amount in each criterion to keep the maximum such loss "low." In particular, if y^* grows at least as fast as $me^{-\mathrm{O}(B)}$, then the output value here is $O(y^*)$, while maintaining $x_j = O(d_j)$ for all j. (Also, if the CIP is uncapacitated, then the above is a significant improvement if B is large.)

We see from (1.2) that in the case where $\ln^+(mB/y^*) \leq B$, both $(c^T \cdot x)/y^*$ and the maximum "violation" $\max_j\ x_j/d_j$ are bounded by constants, which is reasonable. Thus, we consider the case where $\ln^+(mB/y^*) > B$. Here, however, the violation $\max_j\ x_j/d_j$ can be as high as $1 + a_1 \ln^+(mB/y^*)/B$, which is unsatisfactory. If it is not feasible (e.g., for capacity/fault-tolerance reasons) to deviate from the local constraints by this much, then even the gain in the global criterion (caused by the large value of B) will not help justify such a result. Therefore, a natural question is: is it possible to lose a small amount in the global criterion, while losing much less in the local criterion (i.e., in $\max_j\ x_j/d_j$), in the case where $\ln^+(mB/y^*) > B$? We answer this in the affirmative.

[1] Recall that $\ln^+(x)$ denotes $\max\{\ln(x), 1\}$. To parse the term "$\ln^+(mB/y^*)/B$", note that it is $\ln(mB/y^*)/B$ if $y^* \leq me^{-B}$ and is $O(1)$ otherwise.

(a) For the important special case of *unweighted* CIPs (for all j, $c_j = 1$), consider the case $\ln^+(mB/y^*) > B$. Then, for any parameter ϵ, $0 < \epsilon < 1$, we present an algorithm that outputs an x with

(i) $x_j \leq \lceil d_j/(1-\epsilon) \rceil$ for all j, where

(ii) the objective function value is at most $a_2 y^*(1/(1-\epsilon) + (1/\epsilon^2)\ln^+(mB/y^*)/B)$ for an absolute constant $a_2 > 0$.

Note the significant improvement over (1.1) and (1.2), particularly if ϵ is a constant: by losing just a constant factor in the output value of the objective function, we have ensured that each x_j/d_j is bounded by a *constant* (at most $1/(1-\epsilon) + 1/d_j$). This is an improvement over the bound stated in (1.2). In our view, ensuring little loss in the local criterion here is quite important as it involves *all* the variables x_j (e.g., all the nodes of a graph in facility location) and since $\max_j\ x_j/d_j$ may be *required* to be low due to physical and other constraints.

(b) For the case where the coefficient matrix A has only zeroes and ones and where a feasible solution (i.e., for all j, $x_j \leq d_j$) to a (possibly weighted) CIP is really required, we present an approximation algorithm with output value at most $O(y^* \ln^+(m/y^*))$. This works whether $\ln^+(mB/y^*) > B$ or not. While incomparable with the results of [7, 8], this is better if y^* is bigger than a certain threshold. This is also seen to be an improvement over the $O(y^* \lg m)$ bound of [5] if $y^* \geq m^a$, where $a \in (0, 1)$ is an absolute constant.

Thus, this work presents improved local vs. global balancing for a family of problems: the basic packet routing problem (the first constant-factor approximation) and CIPs (gaining more than a constant factor in the local criterion while losing a constant factor in the global criterion). The structure of the rest of the paper is as follows. In section 2, we discuss the algorithm for the packet routing problem, which consists mainly of three steps: (1) constructing and solving an LP relaxation (section 2.1); (2) obtaining a set of routes via suitable rounding (section 2.2); and (3) scheduling the packets (section 2.3) using the algorithm of [17]. The nature of our LP relaxation also provides an interesting re-interpretation of our result, as shown by Theorem 2.4 in section 2.3. We discuss in section 2.4 an extension of our idea to a more general setting, where the routing problem is replaced by a canonical covering problem. In section 3, we discuss our results for the general CIPs. We present our improved local vs. global balancing for unweighted CIPs in section 3.1; the case where $x_j \leq d_j$ is really required for all j is handled in section 3.2 for the case where the coefficient matrix has only zeroes and ones. (Note, for instance, that the coefficient matrix has only zeroes and ones for the facility location problem discussed in section 1.2.)

2. Approximating the routing time to within a constant factor. We refer the reader to the introduction for the definition and motivation for packet routing. Leighton, Maggs, and Rao, in a seminal paper, studied the issue of scheduling the movement of the packets *given the path to be traversed by each packet* [16]. They showed that the packets can be routed in time proportional to the sum of the *congestion* and *dilation* of the paths selected for each packet (see the sentence preceding section 1.2 for the definition of these two parameters). However, they did not address the issue of path selection; one motivation for their work is that paths can plausibly be selected using, e.g., the well-known “random intermediate destinations” idea [33, 34]. However, no general results on path selection, and hence on the time needed for packet routing, were known for *arbitrary* networks G. We address this issue here by studying the path selection problem.

THEOREM 2.1. *There are constants $c', c'' > 0$ such that the following holds. For any packet routing problem on any network, there is a set of paths and a corresponding schedule that can be constructed in polynomial time such that the routing time is at most c' times the optimal. Furthermore, the maximum queue size at each edge is bounded by c''.*

We shall denote any path from s_k to t_k as an (s_k, t_k)-path. Given a (directed) path P, $E(P)$ will denote its set of (directed) edges.

2.1. A linear programming relaxation. Consider any given packet routing problem. Let us consider any feasible solution for it, where packet k is routed on path P_k. Let D denote the *dilation* of the paths selected, i.e., D is the length of a longest path among the P_k. Clearly, the time to route all the packets is bounded below by D. Similarly, let C denote the congestion of the paths selected, i.e., the maximum number of packets that must traverse any single edge during the entire course of the routing. Clearly, C is also a lower bound on the time needed to route the packets.

Let N denote the number of nodes in the network and K the number of packets in the problem. We now present an LP relaxation for the problem; some of the notation used in this relaxation is explained in the following paragraph.

$$\text{(ROUTING)} \min(C + D)/2 \text{ subject to}$$

$$\sum_{k=1}^{K} x_f^k \le C \quad \forall f \in E(G), \tag{2.1}$$

$$\sum_{f \in E(G)} x_f^k \le D \quad \forall k \in \{1, 2, \ldots, K\}, \tag{2.2}$$

$$\mathcal{N}^k x^k = b^k \quad \forall k \in \{1, 2, \ldots, K\}, \tag{2.3}$$

$$0 \le x_f^k \le 1 \quad \forall k \in \{1, 2, \ldots, K\} \ \forall f \in E(G).$$

The vector x above is basically a "fractional flow" in G, where x_f^k denotes the amount of "flow" of packet k on edge $f \in E(G)$. The superscript k merely indexes a packet and *does not* mean a kth power. The constraints "$\mathcal{N}^k x^k = b^k$" model the requirement that for packet k, (i) a total of one unit of flow leaves s_k and reaches t_k, and (ii) at all other nodes, the net inflow of the flow corresponding to packet k, equals the net outflow of the flow corresponding to packet k. For conciseness, we have avoided explicitly writing out this (obvious) set of constraints above. Constraints (2.1) say that the "fractional congestion" on any edge f is at most C. Constraints (2.2) say that the "fractional dilation" $\sum_f x_f^k$ is at most D. This is a somewhat novel way of relaxing path lengths to their fractional counterparts.

It is easy to see that any path-selection scheme for the packets, i.e., any *integral* flow (where all the x_f^k are either 0 or 1) with congestion C and dilation D, satisfies the above system of inequalities. Thus, since C and D are both lower bounds on the length of the routing time for such a path-selection strategy, so is $(C + D)/2$. Hence, the optimum value of the LP is indeed a lower bound on the routing time for a given routing problem: it is indeed a relaxation. Note that the LP has polynomial size since it has "only" $O(Km)$ variables and $O(Km)$ constraints, where m denotes the number of edges in the network. Thus, it can be solved in polynomial time. Let $\{\overline{x}, \overline{C}, \overline{D}\}$ denote an optimal solution to the program. In section 2.2, we will conduct a certain type of "filtering" on $\overline{x}$. Section 2.3 will then construct a path for each packet and then invoke the algorithm of [17] for packet scheduling.

2.2. Path filtering. The main ideas now are to decompose $\overline{x}$ into a set of "flow paths" via the "flow decomposition" approach and then to adapt the ideas in Lin-Vitter [20] to "filter" the flow paths by effectively eliminating all flow paths of length more than $2\overline{D}$.

The reader is referred to section 3.5 of [1] for the well-known flow decomposition approach. This approach efficiently transforms $\overline{x}$ into a set of flow paths that satisfy the following conditions. For each packet k, we get a collection $\mathcal{Q}^k$ of flows along (s_k, t_k)-paths; each $\mathcal{Q}^k$ has at most m paths. Let $P_{k,i}$ denote the ith path in $\mathcal{Q}^k$. $P_{k,i}$ has an associated flow value $z_{k,i} \geq 0$ such that for each k, $\sum_i z_{k,i} = 1$. (In other words, the unit flow from s_k to t_k has been decomposed into a convex combination of (s_k, t_k)-paths.) The total flow on any edge f will be at most $\overline{C}$:

$$\sum_{(k,i): f \in E(P_{k,i})} z_{k,i} = \sum_{k=1}^{K} \overline{x}_f^k \leq \overline{C}; \tag{2.4}$$

the inequality in (2.4) follows from (2.1). Also, let $|P|$ denote the length of (i.e., the number of edges in) a path P. Importantly, the following bound will hold for each k:

$$\sum_i z_{k,i} |P_{k,i}| = \sum_{f \in E(G)} \overline{x}_f^k \leq \overline{D} \tag{2.5}$$

with the inequality following from (2.2).

The main idea now is to "filter" the flow paths so that only paths of length at most $2\overline{D}$ remain. For each k, define

$$g_k = \sum_{i: |P_{k,i}| > 2\overline{D}} z_{k,i}.$$

It is to easy to check via (2.5) that $g_k \leq 1/2$ for each k. Thus, suppose we define new flow values $\{y_{k,i}\}$ as follows for each k: $y_{k,i} = 0$ if $|P_{k,i}| > 2\overline{D}$, and $y_{k,i} = z_{k,i}/(1 - g_k)$ if $|P_{k,i}| \leq 2\overline{D}$. We still have the property that we have a convex combination of flow values: $\sum_i y_{k,i} = 1$. Also, since $g_k \leq 1/2$ for all k, we have $y_{k,i} \leq 2z_{k,i}$ for all k, i. Therefore, (2.4) implies that the total flow on any edge f is at most $2\overline{C}$:

$$\sum_{(k,i): f \in E(P_{k,i})} y_{k,i} \leq 2\overline{C}. \tag{2.6}$$

Most importantly, by setting $y_{k,i} = 0$ for all the "long" paths $P_{k,i}$ (those of length more than $2\overline{D}$), we have ensured that all the flow paths under consideration are of length at most $O(\overline{D})$. We denote the collection of flow paths for packet k by $\mathcal{P}^k$. For ease of exposition, we will also let y_P denote the flow value of any general flow path P.

Remarks. We now point out two other LP relaxations which can be analyzed similarly and which yield slightly better constants in the approximation guarantee.

- It is possible to directly bound path-lengths in the LP relaxation so that filtering need not be applied; one can show that this improves the approximation guarantee somewhat. On the other hand, such an approach leads to a somewhat more complicated relaxation, and furthermore, binary search has to be applied to get the "optimal" path-length. This, in turn, entails potentially $O(\lg N)$ calls to an LP solver, which increases the running time. Thus, there is a trade-off involved between the running time and the quality of approximation.

- In our LP formulation, we could have used a variable W to stand for $\max\{C, D\}$ in place of C and D; the problem would have been to minimize W subject to the fractional congestion and dilation being at most W. Since W is a lower bound on the optimal routing time, this is indeed a relaxation; using our approach with this formulation leads to a slightly better constant in the quality of our approximation. Nevertheless, we have used our approach to make the relationship between C and D explicit.

2.3. Path selection and routing. Note that $\{y_P : P \in \bigcup_{k=1}^{K} \mathcal{P}^k\}$ is a fractional feasible solution to the following set of inequalities:

$$\sum_k \sum_{P\in\mathcal{P}^k,(i,j)\in E(P)} y_P \le 2\overline{C} \quad \forall f \in E(G),$$
$$\sum_{P\in\mathcal{P}^k} y_P = 1 \quad \forall k.$$

To select one path from $\mathcal{P}^k$ for each packet k, we need to modify the above fractional solution to an integral 0-1 solution. To ensure that the congestion does not increase by much, we shall use the following rounding algorithm of [13].

THEOREM 2.2 (see [13]). *Let A be a real valued $r \times s$ matrix and y be a real-valued s-vector. Let b be a real-valued vector such that $Ay = b$ and t be a positive real number such that, in* every column *of A,* (i) *the sum of all the positive entries is at most t* and (ii) *the sum of all the negative entries is at least $-t$. Then we can compute an integral vector $\overline{y}$ such that for every i, either $\overline{y}_i = \lfloor y_i \rfloor$ or $\overline{y}_i = \lceil y_i \rceil$ and $A\overline{y} = \overline{b}$ where $\overline{b}_i - b_i < t$ for all i. Furthermore, if y contains d nonzero components, the integral approximation can be obtained in time $O(r^3 \lg(1 + s/r) + r^3 + d^2 r + sr)$.*

To use Theorem 2.2, we first transform our linear system above to an equivalent system

$$\sum_k \sum_{P\in\mathcal{P}^k,(i,j)\in E(P)} y_P \le 2\overline{C} \quad \forall\ (i,j) \in E(G),$$
$$\sum_{P\in\mathcal{P}^k} (-2\overline{D}) y_P = -2\overline{D} \quad \forall\ k.$$

The set of variables above is $\{y_P : P \in \bigcup_{k=1}^{K} \mathcal{P}^k\}$. Note that $y_P \in [0, 1]$ for all these variables. Furthermore, in this linear system, the positive column sum is bounded by the maximum length of the paths in $\mathcal{P}^1 \cup \mathcal{P}^2 \cup \cdots \cup \mathcal{P}^K$. Since each path in any $\mathcal{P}^k$ is of length at most $2\overline{D}$ due to our filtering, each positive column sum is at most $2\overline{D}$. Each negative column sum is clearly $-2\overline{D}$. Thus, the parameter t for this linear system, in the notation of Theorem 2.2, can be taken to be $2\overline{D}$. Hence by Theorem 2.2, we can obtain in polynomial time an *integral* solution $\overline{y}$ satisfying

$$\sum_k \sum_{P\in\mathcal{P}^k, f\in E(P)} \overline{y}_P \le 2\overline{C} + 2\overline{D} \quad \forall\ f \in E(G), \tag{2.7}$$

$$\sum_{P\in\mathcal{P}^k} (-2\overline{D}) \overline{y}_P < 0 \quad \forall\ k, \tag{2.8}$$

$$\overline{y}_P \in \{0, 1\} \quad \forall\ P \in \bigcup_{k=1}^{K} \mathcal{P}^k. \tag{2.9}$$

For each packet k, by conditions (2.8) and (2.9), we have $\sum_{P \in \mathcal{P}^k} \overline{y}_P \geq 1$. (Note the crucial role of the *strict* inequality in (2.8).) Thus, for each packet k, we have selected at least one path from s_k to t_k with length at most $2\overline{D}$; furthermore, the congestion is bounded by $2\overline{C}+2\overline{D}$ (from (2.7)). If there are two or more such (s_k, t_k)-paths, we can arbitrarily choose one among them, which of course cannot increase the congestion. The next step is to schedule the packets, given the set of paths selected for each packet. To this end, we use the following result of [17], which provides an algorithm for the existential result of [16].

THEOREM 2.3 (see [17]). *For any set of packets with edge-simple paths having congestion c and dilation d, a routing schedule having length $O(c+d)$ and constant maximum queue size can be found in random polynomial time.*

Applying this theorem to the paths selected from the previous stage, which have congestion $c \leq 2\overline{C} + 2\overline{D}$ and dilation $d \leq 2\overline{D}$, we can route the packets in time $O(\overline{C} + \overline{D})$. Recall that $(\overline{C} + \overline{D})/2$ is a lower bound on the length of the optimal schedule. Thus, we have presented a constant-factor approximation algorithm for the off-line packet routing problem; furthermore, the queue sizes are also bounded by an absolute constant in the routing schedule produced. An interesting related point is that our LP relaxation is reasonable: its integrality gap (worst-case ratio between the optima of the integral and fractional versions) is bounded above by $O(1)$.

An alternative view. There is an equivalent interesting interpretation of Theorem 2.1.

THEOREM 2.4. *Suppose we have an arbitrary routing problem on an arbitrary graph $G = (V, E)$; let L be any nonnegative parameter (e.g., $O(1)$, $O(\lg n)$, $O(\sqrt{n})$). Let $\{(s_k, t_k) : 1 \leq k \leq K\}$ be the set of source-destination pairs for the packets. Suppose we can construct a probability distribution D_k on the (s_k, t_k)-paths for each k such that if we sample, for each packet k, an (s_k, t_k)-path from D_k independently of the other packets, then we have (a) for any edge $e \in E(G)$, the expected congestion on e is at most L, and (b) for each k, the expected length of the (s_k, t_k)-path chosen is at most L. Then, there is a choice of paths for each packet such that the congestion and dilation are $O(L)$. Thus, the routing can be accomplished in $O(L)$ time using constant-sized queues; such a routing can also be constructed off-line in time polynomial in $|V|$ and K.*

We remark that the converse of Theorem 2.4 is trivially true: if an $O(L)$ time routing can be accomplished, we simply let D_k place all the probability on the (s_k, t_k)-path used in such a routing.

Proof of Theorem 2.4. Let π_P^k denote the probability measure of any (s_k, t_k)-path P under the distribution D_k. Let $supp(D_k)$ denote the *support* of D_k, i.e., the set of (s_k, t_k)-paths on which D_k places nonzero probability. The proof follows from the fact that for any $(i, j) \in E(G)$,

$$x_{i,j}^k \equiv \sum_{P:(i,j)\in E(P),\ P \in supp(D_k)} \pi_P^k$$

is a feasible solution to (ROUTING), with C, D replaced by L. Hence, by our filter-round approach, we can construct one path for each packet k such that the congestion and dilation are $O(L)$. As seen above, the path selection and routing strategies can be found in polynomial time. ∎

We consider the above interesting because many fault-tolerance algorithms use very involved ideas to construct a suitable (s_k, t_k)-path for (most) packets [6]. These paths will need to simultaneously have small lengths and lead to small edge congestion.

Theorem 2.4 shows that much more relaxed approaches could work: a distribution that is "good" in expectation on individual elements (edges, paths) is sufficient. Recall that in many "discrete ham-sandwich theorems" (Beck and Spencer [4], Raghavan and Thompson [27]), it is easy to ensure good expectation on individual entities (e.g., the constraints of an integer program), but it is much more difficult to construct one solution that is simultaneously good on all these entities. Our result shows one natural situation where there is just a constant-factor loss in the process.

2.4. Extensions. The result above showing a constant integrality gap for packet routing can be extended to a general family of combinatorial packing problems as follows. Let S_k be the family of all the subsets of vertices S such that $s_k \in S$ and $t_k \notin S$. Recall that the (s_k, t_k)-shortest path problem can be solved as an LP via the following covering formulation:

$$\min \sum_{i,j} c_{i,j} x^k_{i,j} \text{ subject to}$$

$$\sum_{(i,j)\in E:\ i\in S, j\notin S} x^k_{i,j} \geq 1 \quad \forall S \in S_k, \tag{2.10}$$

$$x^k_{i,j} \geq 0 \quad \forall\ (i,j) \in E(G).$$

Constraint (2.10) expresses the idea that "flow" crossing each s-t cut is at least 1. The following is an alternative relaxation for the packet routing problem:

$$\text{(ROUTING-II)} \quad \min(C+D)/2 \text{ subject to}$$

$$\sum_{k=1}^{K} x^k_{i,j} \leq C \quad \forall\ (i,j) \in E(G),$$

$$\sum_{i,j} x^k_{i,j} \leq D \quad \forall\ k,$$

$$\sum_{(i,j)\in E:\ i\in S, j\notin S} x^k_{i,j} \geq 1 \quad \forall S \in S_k, \tag{2.11}$$

$$x^k_{i,j} \in [0,1] \quad \forall\ k, i, j.$$

We can use the method outlined in sections 2.1, 2.2, and 2.3 to show that the optimal solution of (ROUTING-II) is within a constant factor of the optimal routing time. A natural question that arises is whether the above conclusion holds for more general combinatorial packing problems. To address this question, we need to present an alternative (polyhedral) perspective of our (path) selection routine. First we recall some standard definitions from polyhedral combinatorics. The reader is referred to [10] for related concepts.

Suppose we are given a finite set $\mathcal{N} = \{1, 2, \ldots, n\}$ and a family $\mathcal{F}$ of subsets of $\mathcal{N}$. For any $S \subseteq \mathcal{N}$, let $\chi_S \in \{0,1\}^n$ denote the incidence vector of S. We shall consider the problem

$$(OPT) \quad \min\{c^T \chi_F : F \in \mathcal{F}\},$$

where $c \in \Re^n_+$ is a weight function on the elements of $\mathcal{N}$.

Definition 2.5 (see [25]). *The blocking clutter of $\mathcal{F}$ is the family $B(\mathcal{F})$, whose members are precisely those $H \subseteq \mathcal{N}$ that satisfy the following:*

P1. *Intersection: $H \cap F \neq \emptyset$ for all $F \in \mathcal{F}$.*

P2. *Minimality: If H' is any proper subset of H, then H' violates property* P1.

A natural LP relaxation for (OPT) is

$$\min\{c^T x : x \in Q\}, \text{ where } Q = \{x^T \chi_H \geq 1 \text{ for all } H \in B(\mathcal{F}),\ x_i \geq 0 \text{ for all } i\}.$$

Q is known as the blocking polyhedron of $\mathcal{F}$. The following result is well known and easy to check:

$$Q \cap Z^n = \{x \in Z^n :\ \exists F \in \mathcal{F} \text{ such that } \ \forall i \in F,\ x_i \geq 1\}.$$

For several classes of clutters (set-systems), it is known that the extreme points of Q are the integral vectors that correspond to incidence vectors of elements in $\mathcal{F}$. By Minkowski's theorem [25], every element in Q can be expressed as a convex combination of the extreme points and extreme rays in Q. For blocking polyhedra, the set of rays is

$$\{x \in \Re^n :\ \forall i,\ x_i \geq 0\}.$$

Suppose we have a generic integer programming problem that is similar to (ROUTING-II), except for the fact that for each k, (2.11) is replaced by the constraint

$$\sum_{i \in H} x_i^k \geq 1 \ \ \forall\ H \in B(\mathcal{F}^k);$$

$\mathcal{F}^k$ can be any clutter that is well-characterized by its blocking polyhedron Q^k (i.e., the extreme points of the blocking polyhedron Q^k are incidence vectors of the elements of the clutter $\mathcal{F}^k$). Thus, we have a generalization of (ROUTING-II):

$$\text{(BLOCK)} \quad \min(C+D)/2 \text{ subject to}$$

$$\sum_{k=1}^{K} x_i^k \leq C \ \ \forall\ i \in \mathcal{N},$$

$$\sum_{i} x_i^k \leq D \ \ \forall\ k, \tag{2.12}$$

$$\sum_{i \in H} x_i^k \geq 1 \ \ \forall\ k \text{ and } \forall\ H \in B(\mathcal{F}^k),$$

$$x_i^k \in \{0,1\} \ \ \forall\ k \text{ and } \forall i \in \mathcal{N}. \tag{2.13}$$

Note that the variables x are now indexed by elements of the set $\mathcal{N}$. In the previously discussed special cases, the elements of $\mathcal{N}$ are edges or pairs of nodes.

The LP relaxation of (BLOCK) replaces the constraint (2.13) by

$$0 \leq x_i^k \leq 1 \ \ \forall\ k = 1, \ldots, K\ \forall i \in \mathcal{N}.$$

THEOREM 2.6. *The optimal integral solution value of (BLOCK) is at most a constant factor times the optimal value of the LP relaxation.*

Proof. Let $(\overline{x}_i^k : k = 1, \ldots, K; i \in \mathcal{N})$ denote an optimal solution to the LP relaxation. By Caratheodory's theorem [25], for each fixed k, $(\overline{x}_i^k : i \in \mathcal{N})$ can be expressed as a convex combination of extreme points and extreme rays of the blocking polyhedron Q^k. However, note that the objective function can improve only by decreasing the value of $(\overline{x}_i^k)$ coordinatewise, as long as the solution remains feasible. Furthermore, the extreme rays of the blocking polyhedron correspond to vectors v with each v_i nonnegative. Thus, w.l.o.g., we may assume that the LP optimum is *lexicographically minimal*. This ensures that the optimal solution $(\overline{x}_i^k)$ can be expressed as a convex combination of the extreme points of the polyhedron alone. As seen above, the extreme points in this case are incidence vectors of elements of the kth clutter (we use polyhedral language to let "kth clutter" denote the set-system $\mathcal{F}^k$).

Let $\overline{C}$ and $\overline{D}$ denote the fractional congestion and fractional dilation of the optimal solution obtained by the LP relaxation of (BLOCK). Let $A_1^k, A_2^k, \ldots$ denote incidence vectors of the elements in the kth clutter, and let $A_j^k(i)$ be the ith coordinate of A_j^k. Then we have a convex combination for each k:

$$\forall i,\ \overline{x}_i^k = \sum_j \alpha_j^k \cdot A_j^k(i), \text{ where}$$

$$\alpha_j^k \geq 0\ \forall j, \text{ and } \sum_j \alpha_j^k = 1.$$

Thus, by constraints (2.12), $\sum_{j:|A_j^k|\leq 2\overline{D}} \alpha_j^k \geq 1/2$, since $\sum_j \alpha_j^k |A_j^k| \leq \overline{D}$.

By filtering out those A_j^k with size greater than $2\overline{D}$, we obtain a set of canonical objects for each k, whose sizes are at most $2\overline{D}$. By scaling the α_j^k by a suitable factor, we also obtain a new set of $\overline{\alpha}_j^k$ such that

$$\sum_{j:|A_j^k|\leq 2\overline{D}} \overline{\alpha}_j^k = 1, \quad \overline{\alpha}_j^k \leq 2\alpha_j^k.$$

Using these canonical objects and $\{\overline{\alpha}_j^k\}$ as the input to Theorem 2.2, we obtain a set of objects (one from each clutter) such that the dilation is not more than $2\overline{D}$ and the congestion not more than $2(\overline{C}+\overline{D})$. Hence the solution obtained is at most $O(1)$ times the LP optimum. ∎

Remark. As pointed out by one of the referees, it is not clear whether the lexicographically minimal optimal solution can be constructed in polynomial time. The above result is thus only about the quality of the LP relaxation. It would be nice to find the most general conditions under which the above can be turned into a polynomial-time approximation algorithm.

3. Improved local vs. global balancing for covering.

Coupled with the results of [16, 17], our approximation algorithm for the routing time (a *global* criterion) also simultaneously kept the maximum queue size (a *local capacity* constraint) constant; our approach there implicitly uses the special structure of the cut covering formulation. We now continue the study of such balancing in the context of CIPs. The reader is referred to section 1.2 for the relevant definitions and history of CIPs. In section 3.1, we will show how to approximate the global criterion well without losing much in the "local" constraints $\{x_j \leq d_j\}$. In section 3.2, we present approximation algorithms for a subfamily of CIPs where $x_j \leq d_j$ is *required* for all j. One of the key tools used in sections 3.1 and 3.2 is Theorem 3.3, which builds on an earlier rounding approach (Theorem 3.2) of [31].

3.1. Balancing local with global. The main result of section 3.1 is Corollary 3.5. This result is concerned with unweighted CIPs and the case where $\ln^+(mB/y^*) > B$. In this setting, Corollary 3.5 shows how the local capacity constraints can be violated much less in comparison with the results of [31], while keeping the objective function value within a constant factor of that of [31].

Let $\exp(x)$ denote e^x; given any nonnegative integer k, let $[k]$ denote the set $\{1, 2, \ldots, k\}$. We start by reviewing the Chernoff–Hoeffding bounds in Theorem 3.1. Let $G(\mu, \delta) \doteq (\exp(\delta)/(1+\delta)^{(1+\delta)})^\mu$, $H(\mu, \delta) \doteq \exp(-\mu\delta^2/2)$.

THEOREM 3.1 (see [23]). *Let $X_1, X_2, \ldots, X_\ell$ be* independent *random variables, each taking values in* $[0, 1]$, $R = \sum_{i=1}^{\ell} X_i$, *and* $E[R] = \mu$. *Then, for any* $\delta \geq 0$, $\Pr(R \geq \mu(1+\delta)) \leq G(\mu, \delta)$. *Also, if* $0 \leq \delta \leq 1$, $\Pr(R \leq \mu(1-\delta)) \leq H(\mu, \delta)$.

We shall use the following easy fact:

$$\forall \mu \geq 0\ \forall \delta \in [0, 1],\ G(\mu, \delta) \leq \exp(-\mu\delta^2/3). \tag{3.1}$$

From now on, we will let $\{x_j^* : j \in [n]\}$ be the set of values for the variables in an arbitrary feasible solution to the LP relaxation of the CIP; thus, $0 \leq x_j^* \leq d_j$. (In particular, x^* could be an optimal LP solution.) Let $y^* = c^T \cdot x^*$. Recall that the case where $\ln^+(mB/y^*) \leq B$ is handled well in [31]; thus we shall assume $\ln^+(mB/y^*) > B$. We now summarize the main result of [31] for CIPs as a theorem. A key ingredient of Theorem 3.2 is the FKG inequality [9].

THEOREM 3.2 (see [31]). *For any given CIP, suppose we are given any* $1 \leq \beta < \alpha < \kappa$ *such that*

$$(1 - H(B\alpha, 1 - \beta/\alpha))^m > G(y^*\alpha, \kappa/\alpha - 1) \tag{3.2}$$

holds. Then we can find in deterministic polynomial time a vector $z = (z_1, z_2, \ldots, z_n)$ *of nonnegative* integers *such that* (a) $(Az)_i \geq b_i\beta$ *for each* $i \in [m]$, (b) $\sum_j c_j z_j \leq y^*\kappa$, *and* (c) $z_j \leq \lceil \alpha x_j^* \rceil \leq \lceil \alpha d_j \rceil$ *for each* $j \in [n]$.

The next theorem presents a rounding algorithm by building on Theorem 3.2.

THEOREM 3.3. *There are positive constants* a_3 *and* a_4 *such that the following holds. Given any parameter* ϵ, $0 < \epsilon < 1$, *let* α *be any value such that* $\alpha \geq (a_3/\epsilon^2)\max\{\ln^+(mB/y^*)/B, 1\}$. *Then we can find in deterministic polynomial time a vector* $z = (z_1, z_2, \ldots, z_n)$ *of nonnegative integers such that* (a) $(Az)_i \geq b_i\alpha(1-\epsilon)$ *for each* $i \in [m]$, (b) $c^T \cdot z \leq a_4 y^*\alpha$, *and* (c) $z_j \leq \lceil \alpha x_j^* \rceil \leq \lceil \alpha d_j \rceil$ *for each* $j \in [n]$.

Remark. It will be shown in the proof of Theorem 3.3 that we can choose, for instance, $a_3 = 3$ and $a_4 = 2$. Since there is a trade-off between a_3 and a_4 that can be fine-tuned for particular applications, we have avoided using specific values for a_3 and a_4 in the statement of Theorem 3.3.

The following simple proposition will also be useful.

PROPOSITION 3.4. *If* $0 < x < 1/e$, *then* $1 - x > \exp(-1.25x)$.

Proof of Theorem 3.3. We choose $a_3 = 3$ and $a_4 = 2$. In the notation of Theorem 3.2, we take $\beta = \alpha(1-\epsilon)$ and $\kappa = a_4\alpha$. Our goal is to validate (3.2); by (3.1), it suffices to show that

$$\exp(-y^*\alpha/3) < (1 - \exp(-B\alpha\epsilon^2/2))^m. \tag{3.3}$$

Note that the left- and right-hand sides of (3.3), respectively, decrease and increase with increasing α; thus, since $\alpha \geq \alpha_0 \doteq (3/\epsilon^2)\max\{\ln^+(mB/y^*)/B, 1\}$ it is enough to prove (3.3) for $\alpha = \alpha_0$. We consider two cases.

Case I. $\ln^+(mB/y^*) \leq B$.

Thus, $\alpha_0 = 3/\epsilon^2$ here. Since $B \geq 1$, we have $\exp(-B\alpha_0\epsilon^2/2) < 1/e$. Therefore, Proposition 3.4 implies that in order to prove (3.3), it suffices to show that

$$y^*\alpha_0/3 \geq 1.25m\exp(-B\alpha_0\epsilon^2/2),$$

i.e., that $y^*/\epsilon^2 \geq 1.25m\exp(-1.5B)$. This is true from the facts that (i) $m/y^* \leq \exp(B)$ (which follows from the fact that $\ln(m/y^*) \leq \ln^+(mB/y^*) \leq B$), and (ii) $\exp(1.5B) \geq \sqrt{e}\exp(B) \geq 1.25\epsilon^2\exp(B)$.

Case II. $\ln^+(mB/y^*) > B$.

Here, it suffices to show that

$$\exp(-y^*\ln^+(mB/y^*)/(B\epsilon^2)) < (1 - \exp(-1.5\cdot\ln^+(mB/y^*)))^m. \tag{3.4}$$

Recall that $\ln^+(mB/y^*) > B \geq 1$. Therefore, we have $mB/y^* > e$, i.e., $y^*/(mB) < 1/e$. Thus,

$$(1 - \exp(-1.5\cdot\ln^+(mB/y^*)))^m = \left(1 - \left(\frac{y^*}{mB}\right)^{1.5}\right)^m > \exp(-1.25m(y^*/(mB))^{1.5}).$$

The inequality follows from Proposition 3.4. Therefore, to establish (3.4), we just need show that

$$y^*\ln^+(mB/y^*)/(B\epsilon^2) \geq 1.25\cdot\sqrt{\frac{y^*}{mB}}\cdot\frac{y^*}{B},$$

i.e., that $\ln^+(mB/y^*)/\epsilon^2 \geq 1.25/\sqrt{e}$, which in turn follows from the facts that $\ln^+(mB/y^*) \geq 1$ and $1/\epsilon^2 \geq 1$. This completes the proof. ∎

Our required result is as follows.

COROLLARY 3.5. *Given any* unweighted *CIP with* $\ln^+(mB/y^*) > B$ *and any parameter* ϵ, $0 < \epsilon < 1$, *we can find in deterministic polynomial time a vector* $v = (v_1, v_2, \ldots, v_n)$ *of nonnegative integers such that* (a) $Av \geq b$, (b) $\sum_j v_j \leq a_2 y^*(1/(1-\epsilon) + (1/\epsilon^2)\ln^+(mB/y^*)/B)$, *where* $a_2 > 0$ *is an absolute constant, and* (c) $v_j \leq \lceil d_j/(1-\epsilon)\rceil$ *for all* j.

Proof. Let $\alpha = \lceil (a_3/\epsilon^2)\ln^+(mB/y^*)/B\rceil$ and z be as in the statement of Theorem 3.3. Define $v_j = \lceil z_j/(\alpha(1-\epsilon))\rceil$ for each j. Conditions (a) and (c) are easy to check, given Theorem 3.3. Since the z_j's are all nonnegative integers and since the CIP is unweighted ($c_j = 1$ for all j), condition (b) of Theorem 3.3 shows that at most $a_4 y^*\alpha$ of them can be nonzero. Thus, condition (b) follows since $v_j \leq z_j/(\alpha(1-\epsilon)) + 1$ if $z_j > 0$ and since $v_j = 0$ if $z_j = 0$. ∎

As mentioned in section 1, this improves the value of $\max_j \ x_j/d_j$ from $O(\ln^+(mB/y^*)/B)$ [31] to $O(1/(1-\epsilon))$, while keeping $(c^T \cdot x)/y^*$ relatively small at $O((1/\epsilon^2)\cdot\ln^+(mB/y^*)/B)$ (as long as ϵ is a constant bounded away from 1).

3.2. Handling stringent constraints. We now handle the case where the constraints $x_j \leq d_j$ have to be satisfied and where the coefficient matrix A has only zeroes and ones. Recall from section 1 that there is a family of facility location problems where the coefficient matrix has only zeroes and ones; this is an example of the CIPs to which the following results apply.

We start with a technical lemma.

LEMMA 3.6. *For any* $0 = u_0 < u_1 \leq u_2 \leq \cdots \leq u_i$ *and any* $\ell > 0$, *the sum* $s_i = \sum_{j=1}^{i}(u_j - u_{j-1})\ln^+(\ell/u_j)$ *is at most* $u_i\ln^+(\ell/u_i) + u_i$.

Proof. If $u_1 \geq \ell/e$, then $s_i = \sum_{j=1}^{i}(u_j - u_{j-1}) = u_i$. Otherwise, let $r \geq 1$ be the highest index such that $u_r < \ell/e$. Thus, $s_i = t_r + u_i - u_r$, where $t_r = \sum_{j=1}^{r}(u_j - u_{j-1})\ln(\ell/u_j)$. Since

$$\sum_{j=1\dots r}(u_j - u_{j-1})\ln(l/u_j) \leq \int_0^{u_r} \ln(l/x)dx = \left.(x\ln(l/x) + x)\right|_0^{u_r},$$

it follows that $t_r \leq u_r \ln(\ell/u_r) + u_r$. Hence,

$$s_i = t_r + u_i - u_r \leq u_r \ln(\ell/u_r) + u_i \leq u_i \ln^+(\ell/u_i) + u_i;$$

the last inequality follows from the fact that for any $x \leq y$ such that $x < \ell/e$, $x\ln(\ell/x) \leq y\ln^+(\ell/y)$. □

The following simple proposition will also help.

PROPOSITION 3.7. *For any $\ell > 0$ and $\ell' \geq 1$, $\ln^+(\ell) \geq (\ln^+(\ell\ell'))/\ell'$.*

Proof. The proposition is immediate if $\ell\ell' \leq e$. Next note that for any $a \geq e$, the function $g_a(x) = \ln(ax)/x$ decreases as x increases from 1 to infinity. Therefore, if $\ell \leq e$ and $\ell\ell' > e$, then

$$(\ln^+(\ell\ell'))/\ell' = (\ln(\ell\ell'))/\ell' = g_\ell(\ell') \leq g_e(\ell') \leq g_e(1) = 1 = \ln^+(\ell).$$

Finally, if $\ell > e$ and $\ell\ell' > e$, then $(\ln^+(\ell\ell'))/\ell' = g_\ell(\ell') \leq g_\ell(1) = \ln^+(\ell)$. □

THEOREM 3.8. *Suppose we are given a CIP with the matrix A having only zeroes and ones. In deterministic polynomial time, we can construct a feasible solution z to the CIP with $z_j \leq d_j$ for each j, and such that the objective function value $c^T \cdot z$ is $O(y^* \ln^+(m/y^*))$.*

Proof. Let a_3 and a_4 be as in the proof of Theorem 3.3. Define $a_5 = \max\{2, 4a_3\}$ and, for any $S \subseteq [n]$, $y_S^* = \sum_{j\in S} c_j x_j^*$. Starting with $S_0 = [n]$, we construct a sequence of sets $S_0 \supset S_1 \supset \cdots$ as follows. Suppose we have constructed $S_0, S_1, \dots, S_i$ so far. Let $h_i = y_{S_i}^*$. If $S_i = \emptyset$, we stop; or else, if all $j \in S_i$ satisfy $a_5 \ln^+(m/h_i)x_j^* \leq d_j$, we stop. If not, define the proper subset S_{i+1} of S_i to be $\{j \in S_i : \ a_5 \ln^+(m/h_i)x_j^* > d_j\}$. For all $j \in (S_i - S_{i+1})$, we fix z_j to be $d_j \geq x_j^*$: note that for all such j, $z_j \leq a_5 \ln^+(m/h_i)x_j^*$.

Let S_t be the final set we construct. If $S_t = \emptyset$, we do nothing more; since $z_j \geq x_j^*$ for all j, we will have $Az \geq b$ as required. Also, it is easy to check that $z_j \leq d_j$ for all j. Therefore suppose $S_t \neq \emptyset$. Let $\alpha = a_5 \ln^+(m/h_t)$. Since we stopped at the nonempty set S_t, we see that $\alpha x_j^* \leq d_j$ for all $j \in S_t$. Recall that for all $j \notin S_t$, we have fixed the value of z_j to be $d_j \geq x_j^*$. Let w denote the vector of the remaining variables, i.e., the restriction of x^* to S_t. Let A' be the submatrix of A induced by the columns corresponding to S_t. We will now focus on rounding each x_j^* ($j \in S_t$) to a suitable nonnegative integer $z_j \leq d_j$.

Define, for each $i \in [m]$,

$$b'_i = b_i - \sum_{j\notin S_t} A_{i,j} z_j;$$

since $z_j \geq x_j^*$ for all $j \notin S_t$, we get

$$(A'w)_i = \sum_{j\in S_t} A_{i,j} x_j^* \geq b'_i \ \forall i \in [m].$$

Since each b_i and $A_{i,j}$ is an integer, so is each b'_i. Suppose $b'_i \leq 0$ for some i. Then, whatever nonnegative integers z_j we round the $j \in S_t$ to, we will satisfy the constraint $(Az)_i \geq b_i$. Therefore, we can ignore such indices i and assume w.l.o.g. that $B' \doteq \min_i b'_i \geq 1$. (The constraints corresponding to indices i with $b'_i \leq 0$ can be retained as "dummy constraints.") Define $\epsilon = 1/2$; recall that $\alpha = a_5 \ln^+(m/h_t)$. Therefore Proposition 3.7 shows that

$$\alpha \geq 4a_3 \max\{\ln^+(mB'/h_t)/B', 1\},$$

i.e., that $\alpha \geq (a_3/\epsilon^2) \max\{\ln^+(mB'/h_t)/B', 1\}$. Thus, by Theorem 3.3, we can round each x^*_j $(j \in S_t)$ to some nonnegative integer $z_j \leq \lceil \alpha x^*_j \rceil \leq d_j$ in such a manner that

$$\sum_{j \in S_t} c_j z_j = O(h_t \alpha), \text{ and } \forall i \in [m], (A'z)_i \geq b_i \alpha(1-\epsilon) \geq b_i; \tag{3.5}$$

the last inequality (i.e., that $\alpha(1-\epsilon) \geq 1/2$) follows from the fact that $\alpha \geq a_5 \geq 2$. Therefore we can check that the final solution is indeed feasible. We need only to bound the objective function value, which we proceed to do now.

We first bound

$$\sum_{j \notin S_t} c_j z_j = \sum_{i=0}^{t-1} \sum_{j \in (S_i - S_{i+1})} c_j z_j. \tag{3.6}$$

Fix any i, $0 \leq i \leq t-1$. Recall that for each $j \in (S_i - S_{i+1})$, we set $z_j = d_j \leq a_5 \ln^+(m/h_i) x^*_j$. Thus,

$$\sum_{j \in (S_i - S_{i+1})} c_j z_j \leq O(\ln^+(m/h_i) \sum_{j \in (S_i - S_{i+1})} c_j x^*_j) = O(\ln^+(m/h_i) \cdot (h_i - h_{i+1})). \tag{3.7}$$

Setting $u_i = h_{t+1-i}$ and substituting (3.7) into (3.6),

$$\sum_{j \notin S_t} c_j z_j = O\left(\sum_{i=2}^{t+1} (u_i - u_{i-1}) \ln^+(m/u_i)\right), \tag{3.8}$$

where $u_{t+1} = y^*$. Now, if $S_t = \emptyset$, (3.8) gives the final objective function value. Otherwise, if $S_t \neq \emptyset$, (3.5) shows that

$$\sum_{j \in S_t} c_j z_j = O(h_t \alpha) = O(u_1 \ln^+(m/u_1)).$$

This, in combination with (3.8) and Lemma 3.6, shows that $\sum_j c_j z_j = O(y^* \ln^+(m/y^*))$.

This completes the proof. □

4. Conclusion. In this paper, we analyze various classes of problems in the context of balancing global vs. local criteria.

Our main result is the first constant-factor approximation algorithm for the off-line packet routing problem on arbitrary networks: for certain positive constants c' and c'', we show that given any packet routing problem, the routing time can efficiently be approximated to within a factor of c', while ensuring that all edge-queues are of size at most c''. Our result builds on the work of [16, 17], while exploiting an interesting trade-off between a (hard) congestion criterion and an (easy) dilation criterion. Furthermore, we show that the result can be applied to a more general setting, by providing a polyhedral perspective of our technique. Our approach of appropriately using the rounding theorem of [13] has subsequently been applied by Bar-Noy et al. [3] to develop approximation algorithms for a family of multicasting problems. It has also been applied for a family of routing problems by Andrews and Zhang [2].

The second major result in the paper improves upon a class of results in multicriteria CIPs. We show that the local criterion of unweighted CIPs can be improved from an approximately logarithmic factor to a constant factor with the global criterion not deteriorating by more than a constant factor (i.e., we maintain a logarithmic factor approximation).

The third main result improves upon a well-known bound for CIPs, in the case where the coefficient matrix A has only zeroes and ones. We show that the approximation ratio can be improved from $O(y^* \lg m)$ to $O(y^* \ln^+(m/y^*))$.

Some open questions are as follows. It would be interesting to study our packet routing algorithm empirically and to fine-tune the algorithm based on experimental observation. It would also be useful to determine the best (constant) approximation possible in approximating the routing time. An intriguing open question is whether there is a *distributed* packet routing algorithm with a constant-factor approximation guarantee. Finally, in the context of covering integer programs, can we approximate the objective function to within bounds such as ours, with (essentially) no violation of the local capacity constraints?

Acknowledgments. We thank Bruce Maggs, the STOC 1997 program committee member(s) and referee(s), and the journal referees for their helpful comments. These have helped improve the quality of this paper a great deal. In particular, one of the journal referees simplified our original proof of Lemma 3.6.

REFERENCES

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1993.

[2] M. Andrews and L. Zhang, *Packet routing with arbitrary end-to-end delay requirements*, in Proceedings of the ACM Symposium on Theory of Computing, Atlanta, GA, 1999, pp. 557–565.

[3] A. Bar-Noy, S. Guha, J. (Seffi) Naor, and B. Schieber, *Multicasting in heterogeneous networks*, in Proceedings of the ACM Symposium on Theory of Computing, Dallas, TX, 1998.

[4] J. Beck and J. H. Spencer, *Integral approximation sequences*, Math. Programming, 30 (1984), pp. 88–98.

[5] D. Bertsimas and R. Vohra, *Rounding algorithms for covering problems*, Math. Programming, 80 (1998), pp. 63–89.

[6] R. Cole, B. M. Maggs, and R. Sitaraman, *Routing on butterfly networks with random faults*, in Proceedings of the IEEE Symposium on Foundations of Computer Science, Milwaukee, WI, 1995, pp. 558–570.

[7] G. Dobson, *Worst-case analysis of greedy heuristics for integer programming with nonnegative data*, Math. Oper. Res., 7 (1982), pp. 515–531.

[8] M. L. Fisher and L. A. Wolsey, *On the greedy heuristic for continuous covering and packing problems*, SIAM J. Alg. Disc. Meth., 3 (1982), pp. 584–591.

[9] C. M. Fortuin, J. Ginibre, and P. N. Kasteleyn, *Correlational inequalities for partially ordered sets*, Comm. Math. Phys., 22 (1971), pp. 89–103.

[10] D. R. Fulkerson, *Blocking polyhedra*, in Graph Theory and Its Applications, B. Harris, ed., Academic Press, New York, 1970, pp. 93–112.

[11] L. A. Hall, A. S. Schulz, D. B. Shmoys, and J. Wein, *Scheduling to minimize average completion time: Off-line and on-line approximation algorithms*, Math. Oper. Res., 22 (1997), pp. 513–544.

[12] C. Kaklamanis, A. R. Karlin, F. T. Leighton, V. Milenkovic, P. Raghavan, S. B. Rao, C. Thomborson, and A. Tsantilas, *Asymptotically tight bounds for computing with faulty arrays of processors*, in Proceedings of the IEEE Symposium on Foundations of Computer Science, St. Louis, MO, 1990, pp. 285–296.

[13] R. M. Karp, F. T. Leighton, R. L. Rivest, C. D. Thompson, U. V. Vazirani, and V. V. Vazirani, *Global wire routing in two-dimensional arrays*, Algorithmica, 2 (1987), pp. 113–129.

[14] F. T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*, Morgan Kaufmann, San Mateo, CA, 1992.

[15] F. T. Leighton, *Methods for message routing in parallel machines*, in Proceedings of the ACM Symposium on the Theory of Computing, Victoria, Canada, 1992, pp. 77–96.

[16] F. T. Leighton, B. M. Maggs, and S. B. Rao, *Packet routing and job-shop scheduling in O(congestion+dilation) steps*, Combinatorica, 14 (1994), pp. 167–186.

[17] F. T. Leighton, B. M. Maggs, and A. Richa, *Fast algorithms for finding O(congestion+dilation) packet routing schedules*, Combinatorica, 19 (1999), pp. 375–401.

[18] F. T. Leighton, B. M. Maggs, and R. K. Sitaraman, *On the fault tolerance of some popular bounded-degree networks*, SIAM J. Comput., 27 (1998), pp. 1303–1333.

[19] F. T. Leighton and S. B. Rao, *An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms*, in Proceedings of the IEEE Symposium on Foundations of Computer Science, White Plains, NY, 1988, pp. 422–431.

[20] J. H. Lin and J. S. Vitter, *ϵ-approximations with minimum packing constraint violation*, in Proceedings of the ACM Symposium on Theory of Computing, Victoria, Canada, 1992, pp. 771–782.

[21] M. V. Marathe, R. Ravi, and R. Sundaram, *Service-constrained network design problems*, Nordic J. Comput., 3 (1996), pp. 367–387.

[22] S. T. McCormick and M. L. Pinedo, *Scheduling n independent jobs on m uniform machines with both flow time and makespan objectives: A parametric approach*, ORSA J. Comput., 7 (1992), pp. 63–77.

[23] R. Motwani and P. Raghavan, *Randomized Algorithms*, Cambridge University Press, Cambridge, UK, 1995.

[24] M. Naor and R. M. Roth, *Optimal file sharing in distributed networks*, SIAM J. Comput., 24 (1995), pp. 158–183.

[25] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*, Wiley and Sons, New York, 1988.

[26] R. Ostrovsky and Y. Rabani, *Universal O(congestion + dilation + $\log^{1+\epsilon} N$) local control packet switching algorithms*, in Proceedings of the ACM Symposium on Theory of Computing, El Paso, TX, 1997, pp. 644–653.

[27] P. Raghavan and C. D. Thompson, *Randomized rounding: A technique for provably good algorithms and algorithmic proofs*, Combinatorica, 7 (1987), pp. 365–374.

[28] A. Ranade, *How to emulate shared memory*, J. Comput. System Sci., 41 (1991), pp. 307–326.

[29] R. Ravi, M. V. Marathe, S. S. Ravi, D. J. Rosenkrantz, and H. B. Hunt III, *Many birds with one stone: Multi-objective approximation algorithms*, in Proceedings of the ACM Symposium on Theory of Computing, San Diego, CA, 1993, pp. 438–447.

[30] D. B. Shmoys and É. Tardos, *An approximation algorithm for the generalized assignment problem*, Math. Programming, 62 (1993), pp. 461–474.

[31] A. Srinivasan, *Improved approximation guarantees for packing and covering integer programs*, SIAM J. Comput., 29 (1999), pp. 648–670.

[32] C. Stein and J. Wein, *On the existence of schedules that are near-optimal for both makespan and total weighted completion time*, Oper. Res. Lett., 21 (1997), pp. 115–122.

[33] L. G. Valiant, *A scheme for fast parallel communication*, SIAM J. Comput., 11 (1982), pp. 350–361.

[34] L. G. Valiant and G. J. Brebner, *Universal schemes for parallel communication*, in Proceedings of the ACM Symposium on Theory of Computing, Milwaukee, WI, 1981, pp. 263–277.