Greedy Distributed Optimization of Multi-Commodity Flows

[Extended Abstract]

Baruch Awerbuch*
Johns Hopkins University.
baruch@cs.jhu.edu

Rohit Khandekar
IBM T.J. Watson Research Center.
rkhandekar@gmail.com.

ABSTRACT

The multi-commodity flow problem is a classical combinatorial optimization problem that addresses a number of practically important issues of congestion and bandwidth management in connection-oriented network architectures.

We consider solutions for distributed multi-commodity flow problems, which are solved by multiple agents operating in a cooperative but uncoordinated manner. We provide the first *stateless* greedy distributed algorithm for the concurrent multi-commodity flow problem with poly-logarithmic convergence. More precisely, our algorithm achieves $1+\epsilon$ approximation, with running time $O(\log \mathrm{P} \cdot \log^{O(1)} m \cdot (1/\epsilon)^{O(1)})$ where P is the number of flow-paths in the network. No prior results exist for our model.

Our algorithm is a reasonable alternative to existing polynomial sequential approximation algorithms, such as Garg-Könemann [17]. The algorithm is simple and can be easily implemented or taught in a classroom.

Remarkably, our algorithm requires that the increase in the flow rate on a link is more *aggressive* than the decrease in the rate. Essentially all of the existing flow-control heuristics are variations of TCP, which uses a conservative cap on the increase (e.g., additive), and a rather liberal cap on the decrease (e.g., multiplicative). In contrast, our algorithm requires the increase to be multiplicative, and that this increase is *dramatically more aggressive* than the decrease.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: [Non-numerical Algorithms and Problems]

General Terms

algorithms, theory

*Partially supported by NSF grants CCF 0515080, ANIR-0240551, CCR-0311795, and CNS-0617883.

PODC'07, August 12–15, 2007, Portland, Oregon, USA.

Keywords

multi-commodity flows, distributed algorithms, statelessness, self-stabilization

1. INTRODUCTION

The goal of this paper is to optimize resource allocation in a decentralized network architecture (say, the next generation Internet) where different applications compete for shared network resources in a "greedy" manner, without explicit coordination with each other, while being subjected to some regulatory constraints that limit their behavior.

1.1 Stateless algorithms

For a number of reasons, it is desirable that such an optimization is performed in a distributed and "greedy" manner. This means, for example, that the flows make their routing decisions in a cooperative but uncoordinated manner, without having access to a global clock and without being able to properly initialize and synchronize their individual executions. The flows will be only observing the current network congestion, without being able to pin the individual contributions of other commodities, and without keeping any memory about the past. Statelessness is attractive since it implies a number of other important features, which are very desirable in networks and distributed systems with unreliable components:

Self-stabilization. It is a classical and a very elegant notion in the theory of robust distributed systems [12, 19, 13, 9, 8]. It means that the solution can withstand adversarial but finite sequence of "hard reset" events, namely, crashes accompanied with loss of all memory contents, except of course the code of the program to be executed. Note that a self-stabilizing solution allows some of the agents to fall asleep for an undetermined period of time, and then to wake up. Also it means that the algorithm does not need to be initialized.

Incremental and local adjustment. Even if changes occur in the network topology or demand pattern, the algorithm does not need to be restarted. Rather, the algorithm adjusts the flows in a local and incremental manner, without disrupting the flows that are not affected.

No global clock. Algorithms should not be driven by a global clock; however we assume that local clocks generate perfectly synchronized (but un-numbered) *rounds*, without maintaining a global time. Existing path-based flow algorithms [1, 28, 10, 35, 5] crucially depend on maintaining a

state (e.g., proper initialization) which is achieved by maintaining global time, and thus do not work in our model.

Our objective is to replace the current flow control and routing mechanisms, namely the routing metric (e.g., minimum-hops) and the flow control (e.g., TCP) that do not provide any tangible guarantees in terms of optimizing multi-commodity flows. We now describe the main building blocks of a routing algorithm.

Routing metric. Regardless of the original preferences of the users, the "traffic engineering" paradigm means essentially imposing a global cost metric on all the flows. In order to make routing more robust, such metric should be congestion sensitive, and respond dynamically to changing congestion patterns.

Originally, Internet traffic has been congestion-driven, and different flows tried to reroute themselves along least congested paths [30]. The following phenomenon, referred as "tragedy of the commons" has been detected by the practitioners [29]. Since each flow attempts to route over the least congested path and since congestion on the path grows with traffic, the least-congested-path routing is inherently unstable — it leads to unpredictable oscillations and wastes network resources.

In order to avoid the instability problem, Internet routing protocols since the seventies [29] avoid dynamic (congestion caused) rerouting altogether [26, 21, 33, 25], and thus operate over static and potentially significantly sub-optimal paths. Since current Internet routing is congestion-insensitive minimum-hop routing, it is obviously more vulnerable to attacks than an adaptive congestion-adjusted routing proposed in this work. One of the consequences is that current Internet routing architecture does not support any quality of service guarantees and the most basic security guarantees. Today the inability to deal with congestion, in an adaptive manner by rerouting Internet traffic around temporary traffic bottlenecks, remains one of the fundamental unresolved problems in the algorithmic theory of networking.

Flow control mechanism. To remedy the instability problem, the network must impose certain "speed limits" that control the maximal amount of flow increase or decrease on a certain network router. For example, current flow control on Internet is the classic TCP protocol, which allows additive increase for each flow on each edge and multiplicative decrease. Obviously, if there are no more packets to send, the flow does not need to observe the downward speed limit. Flow control is *necessary* since otherwise concurrent operation with greedy users certainly leads to a collapse.

1.2 Organization of this extended abstract

Sec. 2 states the model, Sec. 3 states our results. Sec. 4 reviews existing work. Sec. 5 presents our algorithm. The proofs are organized as follows. Sec. 6 presents our definition and proof of near-optimality of aggregate equilibria. Sec. 7 and 8 analyze flow dynamics and fast convergence to equilibrium. Finally, we conclude with some open questions in Sec. 9.

2. THE STATEMENT OF THE PROBLEM

The concurrent multi-commodity flow problem. Consider a directed capacitated graph (V, E, c) where $c : E \to \Re^+$ is a capacity assignment to the edges. There is a set of commodities $\mathcal{U}$, each commodity i associated with a source $s_i \in V$, a sink $t_i \in V$, and a demand $d_i > 0$. We are also given an upper bound H on the number of edges allowed on any flow-path carrying a positive flow. A flow assignment $\mathbf{f}_i : E \to \Re^+$ for each commodity i is called feasible if it satisfies flow-conservation at all nodes except s_i and t_i, routes a total of d_i flow between s_i and t_i, and can be decomposed into flow-paths each with at most H edges. Given a flow assignment for each commodity $\mathbf{f} = \{\mathbf{f}_i\}$, the congestion on edge e is defined as

$$\ell(e) = \frac{\mathbf{f}(e)}{c(e)} = \frac{\sum_{i \in \mathcal{U}} \mathbf{f}_i(e)}{c(e)}.$$

The maximum congestion is denoted by $|\mathbf{f}| = \max_{e \in E} \ell(e)$. Our goal is to find a collection of feasible flow assignments $\mathbf{f} = \{\mathbf{f}_i\}$ such that $|\mathbf{f}|$ is minimized.

Billboard computing model. We use the following simplified model of distributed computation. We assume that each commodity is associated with an agent. Each agent is assumed to have an access to a global clock, be oblivious to the existence of other agents, and be not allowed to communicate explicitly with other agents. Each edge is assumed to maintain a "billboard" that records its current total congestion. In each round of the global clock, the agents concurrently "read" the congestion values at different edges from the billboards, decide how to reroute their flows, and subsequently concurrently "write" their changes to these billboards; these changes will be observable in the next round.

Each packet sent by the source carries the full description of the path, and routers just forward the packet to the next hop. Such routing algorithms are also called "Link-State", because they maintain at the end-point (source) of the flow, the database of all the network links, *including* current utilization of those links. Maintaining local link-state databases at the flow sources means that utilization of all the links must be continuously flooded through the network, resulting in larger (but still acceptable) maintenance cost.

In absence of a global clock, it is impossible to reset the protocol at certain predetermined global times. The agents cannot even use time-outs to pass information to other agents. Thus, it is impossible to solve the problem by a centralized algorithm as in [32, 17] or by statefull distributed algorithm [5] that requires initialization. In fact it is not even a-priori obvious that near-optimal distributed stateless solutions even *exist*.

Complexity measures. We will use the following measures to evaluate our algorithms:

- *Approximation ratio* is the ratio between the current performance and the optimal performance.
- *Distributed convergence time* (for a certain approximation ratio) is the number of rounds it takes for the actual flow sent by the agents to start meeting the approximation bounds.
- *Computational complexity* of each agent is the computational overhead imposed by the local program of each agent, in terms of the number of local computational steps.

(Note: The distributed convergence time is an information-theoretic rather than computational measure.)

NOTATION 2.1. *Let m, n, k be the number of edges, vertices, and commodities in the network graph respectively; let H and $\mathrm{P} = n^H$ be the upper bounds on the number of edges on a feasible flow-path and the number of flow-paths, respectively. A flow algorithm is said to have a* poly-logarithmic convergence *if its convergence time is $O(\log \mathrm{P} \cdot \log^{O(1)} m)$.*

Note that $O(\log \mathrm{P}) = O(H \cdot \log n)$. In practice, $n \gg H$; only short paths (less than 10 hops) are used in existing networks. One particularly compelling application is that of clients concurrently attempting to find minimally congested servers. This corresponds to path length $H = 1$, i.e., routing in a bipartite graph. In summary, the convergence in time proportional to path length is a reasonable result in practice.

3. OUR RESULTS AND TECHNIQUES

We provide the *first known* stateless distributed optimization algorithm for the concurrent multi-commodity flow problem with *poly-logarithmic convergence*.

THEOREM 3.1. *Our algorithm (see Section 5, Figures 1,2, 3, 4) achieves $1 + \epsilon$ approximation, with convergence time $O(\log \mathrm{P} \cdot \log^{O(1)} m \cdot (1/\epsilon)^{O(1)})$ for any given $\epsilon > 0$.*

The *computational overhead* is comparable to that of a blocking flow computation; it can be easily upper-bounded by $\tilde{O}\left(H \cdot m^2\right)$ (per commodity). From the point of view of classical sequential computation, our algorithm is only slightly inferior to the $\tilde{O}\left(m^2\right)$-time best existing sequential approximations [17]. The algorithm is simple (approximately a dozen lines of pseudo-code) and is easily implementable.

We associate a routing metric (or cost) with each edge that is exponential in its current congestion. The commodities then continuously reroute their flows to reduce their routing cost, i.e., toward less congested edges. We stress that our algorithm reaches near-optimality *without* ever reaching even approximate Nash equilibrium in the routing metric. In fact, we prove that for every convex cost function, e.g., exponential, near-optimality is implied by a *much weaker* notion of *aggregate equilibrium*, which only requires that the *average* local imbalance is small, with the average being weighed by the relative contributions of the commodities to the total cost.

First we observe that if each flow manages to route itself along the shortest path in the exponential routing metric, then we reach a near-optimum congestion. This can be easily proved using convexity arguments (Section 6.1). However we cannot *simultaneously* route all the flows along the shortest paths, since this would cause congestions on these paths and they would not be the shortest any more. The congestion can be avoided by sending a tiny fraction (inversely proportional to number of commodities) of the demand of each commodity, i.e., using a tiny *additive* speed limit on the way we change each flow. However, this results in a convergence time that is linear in the number of commodities, similar to the sequential flow algorithms such as [22, 23, 27, 32]. The latter work proves convergence under the assumption that flows proceed *sequentially* routing small fraction of their demand along short paths. These proofs are not useful here since they require commodities to be serialized resulting in very large convergence times; our goal is to achieve convergence times that are logarithmic in the number of network paths or linear in the path length (recall that the path length is rather small in practice).

It is essentially impossible to achieve this type of convergence without allowing constant fraction of the demand to be rerouted. Obviously, this is impossible to achieve right away, since this will cause huge congestion and oscillations. However, this is possible to achieve *over time*, by using *multiplicative* speed limits, that enable some fraction of the flow to be increased in each step. Over a certain time interval, this allows to exponentially increase ("pump up") flow into the low-cost areas, and exponentially decrease flow in other areas.

Note that the distributed primal-dual algorithms such as [5] succeed in accomplishing this "pumping up" effect *using an initial state*, and, in particular, using the fact that all flows are monotonically increasing. The major problem in our case is that the continuous rerouting of flows causes exponential routing costs on the edges to be *non-monotonic*, thus preventing us from using techniques of [5, 17]. In [15], a proof of concurrent potential reduction in non-monotonic case is accomplished for the special case of a single commodity. Handling multiple commodities (as in this paper) is much more complex.

One of the most surprising algorithmic ideas is that non-monotonic behavior of flows can be "counter-acted" by enforcing a *much more restrictive* speed limit on the way down than on the way up, which is absolutely opposite of the intuitive perception (e.g., in TCP) that increase is more "dangerous" than decrease. As a result of this idea, we can argue that either cheap paths used by the optimum solution get saturated upward, or large fraction of the total cost (potential) gets reduced. Since the downward speed limit is less aggressive, the "downward" drift cannot counter-act the "upward" drift. Hence the flows of these paths increases exponentially, and a path cannot "survive" for more than logarithmic time. Thus, a significant potential drop must occur within a logarithmic window of time. This "intuition" is correct only if the optimum paths are consistently cheaper than the flow-paths chosen by the algorithm. To deal with fluctuations, we show that if the average cost of the dynamic flow-paths used by a commodity fluctuates, or if the average cost of the cheap paths used by the optimum fluctuate, then the contribution of this commodity to the potential is reduced significantly.

In the analysis of the equilibrium (Section 6) we partition the set of commodities into two sets. The "in-equilibrium" commodities are the ones whose routing cost under the current routing metric is comparable to that of their optimal flow under the current metric. The remaining commodities are "out-of-equilibrium". We show that if majority of the routing cost is due to in-equilibrium commodities, the solution is near-optimal. If on the other hand, significant fraction of the routing cost is due to out-of-equilibrium commodities, then the above argument about potential coming down by a constant factor in logarithmic number of steps holds.

Another important feature that is crucial in proving a logarithmic convergence is using the scaling of the routing metric exponent that depends on the current maximum congestion. This is a significant deviation from the previous

primal-dual exponential-potential-based algorithms [17, 35, 5]. Intuitively, the problems in working with a fixed scaling are: if the scaling factor is too small, a bad initial state with large congestion causes the exponent to become too large to reduce to near optimal in logarithmic number of steps; if, on the other hand, it is too large, the routing metric is not "sensitive" enough to the congestion that it cannot achieve $(1+\epsilon)$ approximation. This technique is in fact reminiscent of the algorithm of Plotkin et al. [32].

4. EXISTING RESULTS

A comparison of our result with existing work is summarized in the discussion below.

Special cases of stateless algorithms in our model. Past work on distributed stateless routing and analysis of routing dynamics, pretty much like in the current paper, does exist, but only provides partial results that work for special cases. The closest work to ours are recent results by Even-Dar et al. [14] and Fisher et al. [16, 15] who state results comparable to our paper in case that commodities operate in a *complete* network (clique). It appears that [15] may be able to handle a general network topology with a common source and sink, which is essentially a single-commodity flow problem. One of the disadvantages of the algorithm in [16, 15] is that its computational overhead is inherently *exponential* (a distinct "agent" needs to be assigned to each path). An open problem stated in [15] is to reduce the computational overhead to polynomial and generalizing to multiple commodities. It is worth to point out that [16, 15] use a more restrictive model where only partial information is available.

Equilibrium/economics approaches. A popular approach in economics, and more recently, in theoretical computer science [24, 34, 2, 3, 11, 31] is to focus on the issue of quality of Nash equilibria, e.g., analyzing "price of anarchy". If each agent is responsible for an infinitesimal amount of flow, then the rerouting along shorter paths decreases the natural potential function, thus causing (eventual) convergence to (approximate) Nash equilibrium, potentially after a long time. Recent work [11, 31] presents *polynomial* bounds on convergence to equilibria. Essentially all of this work uses *polynomial* objective functions; while in order to minimize network congestion one needs to use *exponential* cost function, as in our paper. Note that for our purpose of reaching optimum, approximate equilibrium notion used in [11, 31] is an *overkill* since, as we point out, a weaker concept of *aggregate* equilibrium is sufficient. More importantly, as pointed out by Fisher et al. [16, 15], this research direction does not capture essential and formidable algorithmic obstacles arising in real networks due to decentralized and *concurrent* operation of network flows (e.g., flow control issues on the Internet). One possible interpretation of the Nash equilibrium framework is envisioning an idealized network model where all reroutings occur sequentially, e.g., coordinated by a central server (this is unimaginable for large networks such as the Internet). While such models are certainly of large theoretical interest, they do not lead to stateless algorithms that quickly improve network performance in a concurrent environment.

Statefull algorithms. In a distributed setting, efficient "primal-dual" algorithms in statefull model have been widely studied recently [18, 28, 10, 35, 5, 4]. The initial work in this area [18, 28, 10, 35] used positive Linear Programming approach and thus suffers from exponential representation issue. Note that polynomial representations of MCF problem involve negative coefficients, e.g., to capture Kirkhoff's flow preservation laws and thus do not fit into this framework. This problem has been fixed in the work of Awerbuch, Khandekar and Rao [5], who provided first statefull algorithm with *polynomial* computational overhead and poly-log distributed convergence. Some of these algorithms, e.g., [5, 4] are *poly-logarithmically more efficient* than our algorithm. It is crucial to note that these statefull solutions *fail* in our stateless model since these algorithms need to be *consistently initialized*.

5. THE ALGORITHM

5.1 Greedy concurrent rerouting framework

This section captures much of the flavor of practical routing and flow control mechanisms run on the Internet, as explained in the introduction. Our algorithm has the following components:

Routing metric is an exponential function of the current congestion indicating how the "cost" of an edge grows with its congestion. This metric is imposed to encourage the agents to reroute their flows toward less congested paths.

Flow control rules are created to prevent oscillations due to concurrency:

- The *upward and downward "residual capacities" of an edge* mean that the cumulative effect of all elementary augmentations should not cause too much increase/decrease of flow of each commodity over each edge.
- The *inertia threshold* is the minimal "profit margin" for cost improvement justifying rerouting.

Greedy rerouting means that the agents continue rerouting, or in other words, *circulating* flow from expensive paths to cheaper paths under the routing metric, while observing the flow control rules above. This process continues until the profit margin between unsaturated paths drops below the inertia threshold. The end result of this "maximal" rerouting is then dumped into the network in one atomic step, aggregating all the elementary circulations into one augmenting circulation. This greedy distributed routing framework can be implemented by a stateless algorithm.

5.2 Detailed description of the algorithm

Notations: α is the inertia threshold, ϵ is the approximation guarantee, $\mathbf{f}(e)$ (resp. $\mathbf{f}_i(e)$) is the total flow (resp. flow

1. **Input (read)** flow vectors $\mathbf{f}_i$ and $\mathbf{f}$
2. **Call** RouteMetric, FlowControl, Reroute
3. **Output (write)** flow vector $\mathbf{f}_i$

Figure 1: Procedure Main for commodity i.

Procedure RouteMetric for commodity i

1. $\mu \leftarrow \min\{\mu, 2^{\lfloor \log_2(|\mathbf{f}|\cdot\epsilon)\rfloor}\}$
2. $\forall e \in E$, define $\phi_{e,\mu}(\mathbf{f}(e)) \equiv m^{\frac{\mathbf{f}(e)}{c(e)\cdot\mu}}$
3. $\forall \mathbf{A} \in \mathcal{P}_i$, define $\phi'_\mu(\mathbf{A}) \equiv \sum_{e\in\mathbf{A}} \phi'_{e,\mu}(\mathbf{f}(e))$

Figure 2: Procedure RouteMetric **for commodity** i**.**

Procedure FlowControl for commodity i

1. $\alpha \leftarrow \frac{\epsilon}{40\log m}$
2. $\beta \leftarrow \Theta(\alpha \cdot \frac{\epsilon}{\log m})$
3. $\ddot{\mathbf{f}}_i(e) \leftarrow \frac{\mu}{\log_2 m} \cdot \frac{c(e)}{(1+\beta)k} \cdot \log_2(1+\frac{\alpha}{8})$
4. **upon each round initialize** $\forall e \in E$:
   (a) $\Delta_i^-(e) \leftarrow \mathbf{f}_i(e) \cdot \frac{\beta}{4\cdot H}$
   (b) $\Delta_i^+(e) \leftarrow (1+\beta)\cdot\max\{\mathbf{f}_i(e), \ddot{\mathbf{f}}(e)\} - \mathbf{f}_i(e)$

Figure 3: Procedure FlowControl **for commodity** i**.**

of commodity i) on edge e, $|\mathbf{f}|$ is the maximal congestion observed in the network, μ is initially set to ∞ and is updated to be roughly the largest power to 2 not greater than $(|\mathbf{f}|\cdot\epsilon)$, for a real-valued function ϕ, let ϕ' and ϕ'' denote its first and second derivatives, if ϕ is a function of a real vector, let ϕ' denote its gradient, $\phi'_{e,\mu}(\mathbf{f}(e))$ is the cost of edge e used in the routing metric, $\mathcal{P}_i$ is the set of paths for commodity i, H is maximum number of edges in any path considered in our solution, $\Delta_i^+(e)$ (resp. $\Delta_i^-(e)$) are the upward (resp. downward) residual capacities on edge e.

Main program for commodity i is defined in Figure 1. Upon the beginning of each round, it reads the aggregate flow values and the flow values of commodity i from the billboard, executes the procedures RouteMetric, FlowControl, and Reroute, and subsequently, writes the final flow values of commodity i on the billboard.

Routing metric is defined in Procedure RouteMetric (Figure 2). Note that μ changes in proportion to the maximum congestion $|\mathbf{f}|$. The cost on a path $\mathbf{A}$ is the partial derivative $\phi'_\mu(\mathbf{A})$ (w.r.t. flow sent on that path) of the congestion-dependent potential $\Phi(\mathbf{f}) = \sum_e \phi_{e,\mu}(\mathbf{f}(e))$.

Residual capacity and inertia threshold. Upward ("push") and downward ("pull") residual capacities $\Delta_i^+(e)$ and $\Delta_i^-(e)$ (and parameter β) are defined w.r.t. the flow at the beginning of a round in Procedure FlowControl (Figure 3). Note the crucial multiplicative gap proportional to the maximal path length H between downward and upward residual capacities. If the flow on an edge is very small, we allow an additive increase of up to $(1+\beta)\ddot{\mathbf{f}}(e)$.

Greedy augmentation is given in Procedure Reroute (Figure 4). It *pushes* flow forward over the cheapest path,

Procedure Reroute for commodity i

while $\sum_{v\in V} \Delta_i^-(s_i, v) > 0$ and $\exists\ \mathbf{A}' \in \mathcal{P}_i$ s.t.

- $\min_{e\in\mathbf{A}'} \Delta_i^+(e) > 0$, and
- $d_i \cdot (1+\alpha) \cdot \phi'_\mu(\mathbf{A}') < \sum_{e\in E} \mathbf{f}_i(e)\phi'_{e,\mu}(e)$

do

1. Set $\mathbf{A}^* \leftarrow$ argmin $\phi'_\mu(\mathbf{A}')$ over set of above $\mathbf{A}'$
2. $\delta \leftarrow \min\left\{\min_{e\in\mathbf{A}^*} \Delta_i^+(e), \sum_{v\in V} \Delta_i^-(s_i, v)\right\}$
3. $\forall e \in E,\ \mathbf{f}_i(e) \leftarrow \mathbf{f}_i(e) - \delta \cdot \frac{\mathbf{f}_i(e)}{d_i}$
4. $\forall e \in E,\ \Delta_i^-(e) \leftarrow \Delta_i^-(e) - \delta \cdot \frac{\mathbf{f}_i(e)}{d_i}$
5. $\forall e \in \mathbf{A}^*, \mathbf{f}_i(e) \leftarrow \mathbf{f}_i(e) + \delta$ and $\Delta_i^+(e) \leftarrow \Delta_i^+(e) - \delta$

Figure 4: Procedure Reroute **for commodity** i**.**

and compensates by proportionally *pulling* flow back in *equal proportion* on all the edges — thus creating a circulation with net cost gain — as long as the cheapest cost path is much cheaper than the average cost path, i.e., the flow is "out of equilibrium". The allowed push-forward amount is the minimum upward residual capacity of the cheapest path. The allowed pull-backward amount is the sum of the downward capacities of the source's outgoing edges. If the profit margin is met, the minimum of these two amounts is allowed to circulate, decreasing the residual capacities appropriately.

5.3 Choice of specific parameters

Let $\pi(\ell) = m^{\frac{\ell}{\mu}}$. We refer to any contiguous duration in the algorithm in which the value of μ remains unchanged as a "phase". Consider a phase and let $|\mathbf{f}|_{\max}$ be the maximum congestion on any edge in any round in this phase. Define

$$\vartheta_1 = \min_{0\le\ell\le|\mathbf{f}|_{\max}} \frac{\pi(\ell)}{\pi'(\ell)\cdot\ell}, \qquad \vartheta_2 = \min_{0\le\ell\le|\mathbf{f}|_{\max}} \frac{\pi'(\ell)}{\pi''(\ell)\cdot\ell},$$
$$\text{and} \qquad \vartheta_3 = \min_{0\le\ell\le|\mathbf{f}|_{\max}} \frac{\pi'(\ell)}{\pi''(\ell)\cdot|\mathbf{f}|_{\max}}.$$

Since $|\mathbf{f}|_{\max}$ is not much larger than μ (see Corollary 5.3), from the properties of the exponential function π, it follows that

$$\vartheta_1, \vartheta_2, \vartheta_3 \quad \ge \quad \frac{\mu}{|\mathbf{f}|_{\max}\cdot\log m} \ge \Omega\left(\frac{\epsilon}{\log m}\right).$$

Potential approximation $\aleph_\dagger$**.** Let

$$\Phi(\mathbf{f}) = \sum_e \phi_{e,\mu}(\mathbf{f}(e))$$

be a potential function. Let $\aleph = \frac{\Phi-\Phi^*}{\Phi}$ be the approximation of the potential where Φ^* denotes the potential of the

optimum flow. We choose $\aleph_\dagger < 1$ as the desired approximation of the potential. For our purposes, it is sufficient to choose $\aleph_\dagger = 1/2$. It is easy to see that 1/2-approximation of the potential implies $(1+\epsilon)$-approximation of the maximum congestion.

Approximation threshold α is chosen as $\aleph_\dagger \cdot \vartheta_1/10$.

Multiplicative increase β**.** We define the speed limit β as a parameter such that, for the worst possible online congestion values, $0 \le \ell \le |\mathbf{f}|_{\max}$, the *multiplicative* increase/decrease by a fact $1+\beta$ in the congestion yields at most a factor $1+\alpha/8$ increase/decrease in the cost function π'. Thus we want

$$\pi'(\ell + \ell\beta) < \pi'(\ell) + \pi''(\ell) \cdot 2\ell\beta$$

to be less than $\pi'(\ell) + \pi'(\ell) \cdot \frac{\alpha}{8}$. It is enough to set

$$\beta \;\le\; \frac{\alpha}{16} \cdot \frac{\pi'(\ell)}{\pi''(\ell) \cdot \ell} < \frac{\alpha}{16} \cdot \vartheta_2 = \Theta(\alpha \cdot \vartheta_2)$$

Additive speed limit $\ddot{\mathbf{f}}_i(e)$**.** Note that if the flow of a certain commodity over an edge is small (e.g., zero), the multiplicative increase is ineffective. To bootstrap the increase in the flow, we allow an additive increase $(1+\beta)\ddot{\mathbf{f}}$ in the flow. We choose $\ddot{\mathbf{f}}_i(e)$ so that, the *additive* increase/decrease yields only $\alpha/8$ fraction increase/decrease in the cost $\phi'_{e,\mu}$, i.e.,

$$\ddot{\mathbf{f}}_i(e) \;=\; \frac{\mu}{\log_2 m} \cdot \frac{c(e)}{(1+\beta)k} \cdot \log_2(1+\frac{\alpha}{8}) \qquad (1)$$

For all k commodities, the total additive increase in the flow is at most

$$(1+\beta)k \cdot \ddot{\mathbf{f}}(e) \;\le\; c(e) \cdot \frac{\mu}{\log_2 m} \cdot \log_2(1+\frac{\alpha}{8}) \qquad (2)$$

translating into a multiplicative increase in the cost of

$$m^{\frac{1}{c(e)\cdot\mu} \cdot \frac{c(e)\cdot\mu}{\log_2 m} \cdot \log_2(1+\frac{\alpha}{8})} = 1 + \frac{\alpha}{8}. \qquad (3)$$

5.4 Basic properties

LEMMA 5.1. *The edge-costs $\phi'_{e,\mu}(\mathbf{f}(e))$ increase (resp. decrease) within a round by at most $(1+\alpha/8)^2$ (resp. $(1+\alpha/8)$) factor.*

PROOF. The multiplicative flow increase (resp. decrease) causes the cost to increase (resp. decrease) by at most $1+\alpha/8$ factor. The additive increase can contribute another factor of at most $1+\alpha/8$. □

Comment: Lemma 5.1 shows that the profit margin of α is not "degraded" too much as a result of concurrency. Specifically, the concurrent rerouting leads to a cost change of $(1+\alpha/8)^2$ or $(1+\alpha/8)$ on both sides, i.e., at most $(1+\alpha/2)$ in total. Overall only 1/2 fraction of the profit margin is "degraded". Thus, concurrent rerouting is almost as efficient as sequential rerouting.

DEFINITION 5.1. *Let the* serialization *of the execution be sequential ordering of events in each round, taken in arbitrary order.*

The following two corollaries follow easily.

COROLLARY 5.2. *The profit margin incurred in the* serialization *of the execution is at least $\alpha/2$.*

COROLLARY 5.3. *In a single phase, i.e., while μ is fixed, the potential Φ does not increase and hence $|\mathbf{f}|_{\max}$ is at most $(1+O(\epsilon))$ times the value of $|\mathbf{f}|$ in the beginning of the phase.*

6. AGGREGATE EQUILIBRIA

6.1 Auxiliary potentials

Let $\mathbf{f}$ be the current "online" flow and $\mathbf{g}$ be the optimum "offline" flow. Consider any differentiable convex (e.g., exponential) function $\Phi(\mathbf{f})$ and a convex combination $\mathbf{z} = (1-\eta)\cdot\mathbf{f} + \eta\cdot\mathbf{g}$ where $0 < \eta < 1$. By convexity of Φ, the value at a convex combination $\Phi(\mathbf{z})$ is at most the convex combination of values $(1-\eta)\cdot\Phi(\mathbf{f}) + \eta\cdot\Phi(\mathbf{g})$, i.e.,

$$\Phi\Big((1-\eta)\cdot\mathbf{f} + \eta\cdot\mathbf{g}\Big) \le (1-\eta)\cdot\Phi(\mathbf{f}) + \eta\cdot\Phi(\mathbf{g}).$$

Rearranging we get

$$\Phi\Big(\mathbf{f} - \eta(\mathbf{f}-\mathbf{g})\Big) - \Phi(\mathbf{f}) \le -\eta\Big(\Phi(\mathbf{f}) - \Phi(\mathbf{g})\Big).$$

Dividing by η and taking a limit as $\eta \to 0$, we get

$$-\Phi'(\mathbf{f})\cdot(\mathbf{f}-\mathbf{g}) \le -(\Phi(\mathbf{f}) - \Phi(\mathbf{g})).$$

Thus

$$\Phi(\mathbf{f}) - \Phi(\mathbf{g}) \le \Phi'(\mathbf{f})\cdot\mathbf{f} - \Phi'(\mathbf{f})\cdot\mathbf{g}. \qquad (4)$$

This motivates the following definitions of the "online" and "offline" potentials.

$$\begin{aligned} \Gamma &= \Phi'(\mathbf{f})\cdot\mathbf{f} && \text{(online potential)} \\ \Lambda &= \Phi'(\mathbf{f})\cdot\mathbf{g} && \text{(offline potential)} \end{aligned}$$

These potentials can be subdivided into individual commodities and edges as follows.

$$\begin{aligned} \Gamma_i = \Phi'(\mathbf{f})\cdot\mathbf{f}_i \quad &\text{and} \quad \Lambda_i = \Phi'(\mathbf{f})\cdot\mathbf{g}_i \\ \Gamma(e) = \phi'_{e,\mu}(\mathbf{f}(e))\cdot\mathbf{f}(e) \quad &\text{and} \quad \Lambda(e) = \phi'_{e,\mu}(\mathbf{f}(e))\cdot\mathbf{g}(e) \\ \Gamma_i(e) = \phi'_{e,\mu}(\mathbf{f}(e))\cdot\mathbf{f}_i(e) \quad &\text{and} \quad \Lambda_i(e) = \phi'_{e,\mu}(\mathbf{f}(e))\cdot\mathbf{g}_i(e) \end{aligned}$$

6.2 Aggregate equilibrium

The identity (4) implies that the flow $\mathbf{f}$ is near optimum if the online potentials Γ_i are close to the offline potentials Λ_i for "almost all" commodities.

Let $\Gamma_i(t)$ and $\Lambda_i(t)$ denote the values of Γ_i and Λ_i in round t. Let $\Gamma(t) = \sum_i \Gamma_i(t)$ and $\Lambda(t) = \sum_i \Lambda_i(t)$.

DEFINITION 6.1. *We say that the flows $\mathbf{f}$ are in* aggregate $(\xi,\tilde{\alpha})$-equilibrium *w.r.t. a fixed parameter $\tilde{\alpha}$ at a certain time t if there exists a partition of commodities into equilibrium commodities $\mathcal{C}$ and non-equilibrium commodities $\mathcal{O}$ so that:*

- *The* equilibrium commodities $\mathcal{C}$ *are those for which Λ_i is not much worse than Γ_i:*

$$\mathcal{C} \;=\; \{i \in \mathcal{U} : \Gamma_i(t) \le (1+\tilde{\alpha})\Lambda_i(t)\}.$$

- *The* non-equilibrium *commodities $\mathcal{O}$ are those for which $\Gamma_i(t)$ is much worse than $\Lambda_i(t)$, and the relative contribution of these commodities to $\Gamma(t)$ is less than ξ:*

$$\begin{aligned} \mathcal{O} &= \{i \in \mathcal{U} : \Gamma_i(t) > (1+\tilde{\alpha})\cdot\Lambda_i(t)\}, \\ \xi &> \frac{1}{\Gamma(t)} \cdot \sum_{i\in\mathcal{O}} \Gamma_i(t). \end{aligned}$$

LEMMA 6.1. *If* $\mathbf{f}$ *is in aggregate* $(\xi, \tilde{\alpha})$*-equilibrium where* $\xi \leq \frac{\vartheta_1 \cdot \aleph_\dagger}{2}$ *and* $\tilde{\alpha} \leq 5\alpha$*, then we have achieved the desired approximation* $\aleph_\dagger$*, i.e.,* $\frac{\Phi - \Phi^*}{\Phi} \leq \aleph_\dagger$.

PROOF. Suppose that $\mathbf{f}$ is in aggregate equilibrium. Then we have

$$\begin{aligned}\Gamma - \Lambda &= \sum_{i \in \mathcal{C}} \Phi'(\mathbf{f}) \cdot (\mathbf{f}_i - \mathbf{g}_i) + \sum_{i \in \mathcal{O}} \Phi'(\mathbf{f}) \cdot (\mathbf{f}_i - \mathbf{g}_i) \\ &\leq \sum_{i \in \mathcal{C}} \Phi'(\mathbf{f}) \cdot \mathbf{f}_i \cdot \left(1 - \frac{1}{1+\tilde{\alpha}}\right) + \sum_{i \in \mathcal{O}} \Phi'(\mathbf{f}) \cdot \mathbf{f}_i \\ &\leq \Gamma \cdot \left(\frac{\tilde{\alpha}}{1+\tilde{\alpha}} + \xi\right).\end{aligned}$$

Recalling the definition of ϑ_1, we derive:

$$\begin{aligned}\Phi(\mathbf{f}) &= \sum_e \pi(\ell(e)) \;\geq\; \sum_e \pi'(\ell(e)) \cdot \ell(e) \cdot \vartheta_1 \\ &= \sum_e \frac{\phi'_{e,\mu}(\mathbf{f}(e))}{1/c(e)} \cdot \frac{\mathbf{f}(e)}{c(e)} \cdot \vartheta_1 \;=\; \Gamma \cdot \vartheta_1.\end{aligned}$$

We can now obtain the following bound on the slackness

$$\begin{aligned}\frac{\Phi - \Phi^*}{\Phi} &\leq \frac{\Gamma - \Lambda}{\Phi} \leq \frac{\Gamma}{\Phi}\left(\xi + \frac{\tilde{\alpha}}{1+\tilde{\alpha}}\right) \leq \frac{1}{\vartheta_1}(\xi + \tilde{\alpha}) \\ &\leq \frac{1}{\vartheta_1}\left(\frac{\vartheta_1 \cdot \aleph_\dagger}{2} + 5 \cdot \frac{\aleph_\dagger \cdot \vartheta_1}{10}\right) = \aleph_\dagger.\end{aligned}$$

□

7. DYNAMICS OF POTENTIAL DROP

7.1 Mileage definitions

We define the *online mileage* $\nabla_i^\Gamma(e,t)$, the *offline mileage* $\nabla_i^\Lambda(e,t)$, the *derivative mileage* $\nabla^{\phi'}(e,t)$, as the absolute value of the change, that takes place in round t, in the online potential, offline potential, and derivative of the cost function, respectively.

$$\begin{aligned}\nabla_i^\Gamma(e,t) &= |\Gamma_i(e,t) - \Gamma_i(e,t-1)| \\ \nabla_i^\Lambda(e,t) &= |\Lambda_i(e,t) - \Lambda_i(e,t-1)| \\ \nabla^{\phi'}(e,t) &= |\phi'_{e,\mu}(e,t) - \phi'_{e,\mu}(e,t-1)|\end{aligned}$$

We also define mileage for edges and commodities.

$$\nabla^\Gamma(e,t) = \sum_i \nabla_i^\Gamma(e,t) \quad \text{and} \quad \nabla^\Lambda(e,t) = \sum_i \nabla_i^\Lambda(e,t)$$

$$\nabla_i^\Gamma(t) = \sum_e \nabla_i^\Gamma(e,t) \quad \text{and} \quad \nabla_i^\Lambda(t) = \sum_e \nabla_i^\Lambda(e,t)$$

$$\nabla^\Gamma(t) = \sum_i \nabla_i^\Gamma(t) \quad \text{and} \quad \nabla^\Lambda(t) = \sum_i \nabla_i^\Lambda(t)$$

For an interval $\mathcal{T} = [t_0, t_\ddagger]$, we define

$$\nabla_i^\Gamma(\mathcal{T}) = \sum_{t \in \mathcal{T}} \nabla_i^\Gamma(t) \quad \text{and} \quad \nabla_i^\Lambda(\mathcal{T}) = \sum_{t \in \mathcal{T}} \nabla_i^\Lambda(t).$$

Also for a path $\mathbf{A}$, we define

$$\nabla^{\phi'}(\mathbf{A}, \mathcal{T}) = \sum_{e \in \mathbf{A}} \sum_{t \in \mathcal{T}} \nabla^{\phi'}(e,t).$$

7.2 Mileage-based commodity partitioning

Consider a phase $\mathcal{T} = [t_0, t_\ddagger]$, i.e., the consecutive set of rounds in which the value of μ does not change. We define the online and offline mileage ratios for commodity i as

$$\rho_i^\Gamma(\mathcal{T}) = \frac{\nabla_i^\Gamma(\mathcal{T})}{\Gamma_i(t_0)} \quad \text{and} \quad \rho_i^\Lambda(\mathcal{T}) = \frac{\nabla_i^\Lambda(\mathcal{T})}{\Gamma_i(t_0)}$$

Consider the time t_0 at the beginning of the phase. We now partition the non-equilibrium commodities $\mathcal{O} = \{i \in \mathcal{U} : \Gamma_i(t_0) > (1+\tilde{\alpha}) \cdot \Lambda_i(t_0)\}$ where $\tilde{\alpha} = 5\alpha$ at time t_0 into the following categories: migrant commodities $\mathcal{M}^\Gamma$, mobile commodities $\mathcal{M}^\Lambda$, consistently non-equilibrium commodities $\mathcal{I}$ as follows: $\mathcal{O} = \mathcal{M}^\Gamma \cup \mathcal{M}^\Lambda \cup \mathcal{I}$ where

$$\begin{aligned}\mathcal{M}^\Gamma &= \{i \in \mathcal{O} \mid \rho_i^\Gamma(\mathcal{T}) > \alpha\}, \\ \mathcal{M}^\Lambda &= \{i \in \mathcal{O} \setminus \mathcal{M}^\Gamma \mid \rho_i^\Lambda(\mathcal{T}) > \frac{\alpha^2}{2}\}, \\ \mathcal{I} &= \mathcal{O} \setminus (\mathcal{M}^\Gamma \cup \mathcal{M}^\Lambda).\end{aligned}$$

Let $\xi^\Gamma(t_0), \xi^\Lambda(t_0)$, and $\xi^\mathcal{I}(t_0)$ be the fractions of online potential $\Gamma(t_0)$ at round t_0 for migrant, mobile, and non-equilibrium commodities respectively:

$$\xi^\Gamma = \frac{1}{\Gamma}\sum_{i \in \mathcal{M}^\Gamma} \Gamma_i, \quad \xi^\Lambda = \frac{1}{\Gamma}\sum_{i \in \mathcal{M}^\Lambda} \Gamma_i, \quad \xi^\mathcal{I} = \frac{1}{\Gamma}\sum_{i \in \mathcal{I}} \Gamma_i.$$

Clearly, we have $\xi = \xi^\Gamma + \xi^\Lambda + \xi^\mathcal{I}$.

7.3 Potential drop and length of a phase

To simplify the exposition, we assume that in any round t, the atomic reroutings done by several commodities are serialized. Corollary 5.2 justifies this assumption. Now let $\Delta\Phi(\mathcal{M}^\Gamma)$ (resp. $\Delta\Phi(\mathcal{M}^\Lambda)$ and $\Delta\Phi(\mathcal{M}^\mathcal{I})$) be the total contribution to the potential reduction $\Phi(t_0) - \Phi(t_\ddagger)$ due to the serialized reroutings of commodities in $\mathcal{M}^\Gamma$ (resp. $\mathcal{M}^\Lambda$ and $\mathcal{M}^\mathcal{I}$) in a given duration $[t_0, t_\ddagger]$. Let $\tau_\dagger = \tilde{\Theta}(\frac{H \log^3 m}{\epsilon^2})$.

The following theorem captures the essence of the potential reduction in our algorithm.

THEOREM 7.1. *Consider a phase* $[t_0, t_\ddagger]$*, the following potential improvements take place. Let* $\aleph = \frac{\Phi(t_0) - \Phi^*(t_0)}{\Phi(t_0)}$.

$$\frac{\Delta\Phi(\mathcal{M}^\Gamma)}{\Phi(t_0)} \geq \aleph \cdot \xi^\Gamma(t_0) \cdot \chi^\Gamma \text{ where } \chi^\Gamma = \Omega(\vartheta_2 \cdot \alpha^2) \quad (5)$$

$$\frac{\Delta\Phi(\mathcal{M}^\Lambda)}{\Phi(t_0)} \geq \aleph \cdot \xi^\Lambda(t_0) \cdot \chi^\Lambda \text{ where } \chi^\Lambda = \Omega(\vartheta_3 \cdot \alpha^3) \quad (6)$$

$$\frac{\Delta\Phi(\mathcal{M}^\mathcal{I})}{\Phi(t_0)} \geq \aleph \cdot \xi^\mathcal{I}(t_0) \cdot \chi^\mathcal{I} \text{ where } \chi^\mathcal{I} = \Omega(\alpha) \quad (7)$$

where (7) holds provided $t_\ddagger - t_0 \geq \tau_\dagger = \tilde{\Theta}(\frac{H \log^3 m}{\epsilon^2})$.

The proof of this theorem is outlined in Section 8. Below we note some of its consequences.

THEOREM 7.2. *Consider a phase* $\mathcal{T} = [t_0, t_\ddagger]$ *such that* $\aleph = \frac{\Phi(t) - \Phi^*(t)}{\Phi(t)} > \aleph_\dagger$ *holds at all* $t \in \mathcal{T}$*. Such a phase ends in* $O(\tau_\dagger \cdot \frac{\log^6 m}{\epsilon^6})$ *rounds.*

PROOF. Assume that the phase has at least $\tau_\dagger$ rounds. By Lemma 6.1, either at some time during the phase $\xi^\Gamma + \xi^\Lambda + \xi^\mathcal{I} = \xi \leq \frac{\vartheta_1 \cdot \aleph_\dagger}{2}$, in which case we have already approximated the potential well, namely $\aleph \leq \aleph_\dagger$, or $\xi^\Gamma + \xi^\Lambda + \xi^\mathcal{I} = \xi >$

$\frac{\vartheta_1 \cdot \aleph_\dagger}{2}$ consistently holds during the phase. Assuming the latter holds, we show that phase ends quickly, as required.

Now Theorem 7.1 implies that the overall relative potential drop in an interval of length $\tau_\dagger$ is at least (we drop t_0 and $\mathcal{T}$ from the notation below)

$$\begin{aligned}\frac{\Delta\Phi}{\Phi} &\geq \aleph \cdot (\xi^\Gamma \cdot \chi^\Gamma + \xi^\Lambda \cdot \chi^\Lambda + \xi^\mathcal{I} \cdot \chi^\mathcal{I}) \\ &\geq \aleph_\dagger \cdot (\xi^\Gamma + \xi^\Lambda + \xi^\mathcal{I}) \min\{\chi^\Gamma, \chi^\Lambda, \chi^\mathcal{I}\} \\ &= \frac{1}{2} \cdot \xi \cdot \vartheta_3 \cdot \alpha^3 \geq \frac{1}{2} \cdot \frac{\vartheta_1 \cdot \aleph_\dagger}{2} \cdot \vartheta_3 \cdot \alpha^3 = \Omega\left(\frac{\epsilon^5}{\log^5 m}\right).\end{aligned}$$

Since the initial potential is at most $m^{O(1/\epsilon)}$, after $O(\frac{\log^6 m}{\epsilon^6})$ durations of $\tau_\dagger$ rounds each, the potential reduces significantly causing $|\mathbf{f}|$ to drop by a factor of more than 2. Therefore μ reduces by a factor 2 and the phase ends. □

LEMMA 7.3. *The total number of maximal phases, i.e., maximal durations in which the value of μ remains unchanged, is $O(\log m)$.*

PROOF. Choosing the initial flows to be the maximum flow assignments for each commodity achieves the maximum congestion $|\mathbf{f}| \leq |\mathbf{g}| \cdot O(k) = O(|\mathbf{g}| \cdot m^2)$. Since each phase reduces the maximum congestion by a factor of 2, the proof is complete. □

COROLLARY 7.4. *The algorithm achieves $1+\epsilon$ approximation in $O(\frac{H \log^{O(1)} m}{\epsilon^{O(1)}})$ rounds.*

8. PROVING FAST DROP IN POTENTIAL

Recall that concurrency in the reroutings of different commodities in any round leads to only a constant decrease in potential gain, as shown in Corollary 5.2. So we are going to assume that reroutings are serial.

Fix a commodity i and paths $\mathbf{A}, \mathbf{B} \in \mathcal{P}_i$ with $\phi'_\mu(\mathbf{A}) \geq (1+\Omega(\alpha))\phi'_\mu(\mathbf{B})$. Whenever $\delta\mathbf{f}$ flow is rerouted from $\mathbf{A}$ to $\mathbf{B}$, the drop $\Delta\Phi$ in the overall potential due to this rerouting is roughly $\delta\mathbf{f}$ times the difference of the derivatives.

$$\Delta\Phi \geq \Omega(\delta\mathbf{f}(\phi'_\mu(\mathbf{A}) - \phi'_\mu(\mathbf{B}))) \geq \Omega(\delta\mathbf{f} \cdot \alpha \cdot \phi'_\mu(\mathbf{A}))$$

Note that the above holds even when $\mathbf{A}$ (or $\mathbf{B}$) is not a simple path but is a system of paths carrying certain flow; in this case $\phi'_\mu(\mathbf{A})$ denotes the weighted average of the derivatives for the paths in the system. In what follows, we drop the subscript μ since it is clear from the context.

8.1 Migrant commodities burn potential

We first prove the inequality (5) in Theorem 7.1. For a path $\mathbf{A} \in \mathcal{P}_i$, let $\Gamma(\mathbf{A}) = \sum_{e\in\mathbf{A}} \Gamma(e)$ and let $\Gamma'(\mathbf{A})$ be its derivative w.r.t. the flow on this path. We have

$$\begin{aligned}\Gamma'(\mathbf{A}) &= \sum_{e\in\mathbf{A}} \Gamma'(\mathbf{f}(e)) = \sum_{e\in\mathbf{A}} \left(\phi'_e(\mathbf{f}(e)) + \phi''_e(\mathbf{f}(e)) \cdot \mathbf{f}(e)\right) \\ &= \sum_{e\in\mathbf{A}} \phi'_e(\mathbf{f}(e)) \left(1 + \frac{\pi''(\ell(e)) \cdot \ell(e)}{\pi'(\ell(e))}\right) \\ &\leq \sum_{e\in\mathbf{A}} \phi'_e(\mathbf{f}(e)) \left(1 + \frac{1}{\vartheta_2}\right) = O\left(\phi'(\mathbf{A}) \cdot \frac{1}{\vartheta_2}\right).\end{aligned}$$

Thus we can upper bound the online mileage of commodity i due to rerouting $\delta\mathbf{f}$ flow from $\mathbf{A}$ to $\mathbf{B}$ by the potential reduction due to this rerouting as follows.

$$\begin{aligned}\nabla_i^\Gamma &\leq O(\delta\mathbf{f}(\Gamma'(\mathbf{A}) + \Gamma'(\mathbf{B}))) \leq O\left(\frac{\delta\mathbf{f}}{\vartheta_2}(\phi'(\mathbf{A}) + \phi'(\mathbf{B}))\right) \\ &\leq O\left(\frac{\delta\mathbf{f}}{\vartheta_2}\phi'(\mathbf{A})\right) \leq O\left(\frac{\Delta\Phi}{\vartheta_2 \cdot \alpha}\right).\end{aligned}$$

In other words, we have just proved

$$\Delta\Phi \geq \Omega(\vartheta_2 \cdot \alpha \cdot \nabla_i^\Gamma). \tag{8}$$

Aggregating migrant commodities. For each migrant commodity, each unit of online mileage translates into the potential reduction as given in (8). The total mileage of migrant commodities is

$$\sum_{i\in\mathcal{M}^\Gamma} \rho_i^\Gamma \cdot \Gamma_i(t_0) \geq \alpha \cdot \xi^\Gamma(t_0) \cdot \Gamma(t_0) \geq \alpha \cdot \xi^\Gamma(t_0) \cdot \aleph \cdot \Phi(t_0).$$

This follows since for migrant commodities, $\rho_i^\Gamma \geq \alpha$ and

$$\frac{\Gamma(t_0)}{\Phi(t_0)} \geq \frac{\Phi(t_0) - \Phi^*(t_0)}{\Phi(t_0)} \geq \aleph. \tag{9}$$

Thus the overall potential reduction due to migrant commodities is

$$\Delta\Phi \geq \Omega(\aleph \cdot \xi^\Gamma(t_0) \cdot \vartheta_2 \cdot \alpha^2 \cdot \Phi(t_0)).$$

8.2 Mobile commodities burn potential

We now prove the inequality (6) in Theorem 7.1. For offline potential, we use the following identity.

$$\begin{aligned}\Lambda'(\mathbf{A}) &= \sum_{e\in\mathbf{A}} \Lambda'(\mathbf{f}(e)) = \sum_{e\in\mathbf{A}} \phi''_e(\mathbf{f}(e)) \cdot \mathbf{g}(e) \\ &= \sum_{e\in\mathbf{A}} \phi'_e(\mathbf{f}(e)) \cdot \frac{\pi''(\ell(e)) \cdot \ell^*(e)}{\pi'(\ell(e))} \\ &\leq \sum_{e\in\mathbf{A}} \phi'_e(\mathbf{f}(e)) \cdot \frac{\pi''(\ell(e)) \cdot |\mathbf{f}|_{\max}}{\pi'(\ell(e))} \leq \phi'(\mathbf{A}) \cdot \frac{1}{\vartheta_3}.\end{aligned}$$

Thus the offline mileage can also be amortized against the potential reduction.

$$\begin{aligned}\nabla_i^\Lambda &\leq O(\delta\mathbf{f}(\Lambda'(\mathbf{A}) + \Lambda'(\mathbf{B}))) \leq O\left(\frac{\delta\mathbf{f}}{\vartheta_3}(\phi'(\mathbf{A}) + \phi'(\mathbf{B}))\right) \\ &\leq O\left(\frac{\delta\mathbf{f}}{\vartheta_3}\phi'(\mathbf{A})\right) \leq O\left(\frac{\Delta\Phi}{\vartheta_3 \cdot \alpha}\right).\end{aligned}$$

Thus

$$\Delta\Phi \geq \Omega(\vartheta_3 \cdot \alpha \cdot \nabla_i^\Lambda). \tag{10}$$

Aggregating Mobile commodities. For each one of the mobile commodities, each unit of mileage translates into reduction in potential as in (10). A mobile commodity i has mileage at least $\rho_i^\Lambda \cdot \Gamma_i$. Summing offline mileage over mobile commodities, using equation (9), and using the fact that for mobile commodities $\rho_i^\Lambda \geq \alpha^2/2$, we get

$$\sum_{i\in\mathcal{M}^\Lambda} \rho_i^\Lambda \cdot \Gamma_i(t_0) \geq \frac{\alpha^2}{2}\xi^\Lambda(t_0) \cdot \Gamma(t_0) > \frac{\alpha^2}{2} \cdot \xi^\Lambda(t_0) \cdot \aleph \cdot \Phi(t_0).$$

Overall, mobile commodities reduce potential by at least

$$\Delta\Phi \geq \Omega(\aleph \cdot \xi^\Lambda(t_0) \cdot \vartheta_3 \cdot \alpha^3 \cdot \Phi(t_0)).$$

8.3 Contribution of non-equilibrium commodities

Now we prove (7) of Theorem 7.1.

LEMMA 8.1. *Consider a sample space Ω and a probability function $\pi : \Omega \to \Re^+$. Furthermore, consider two functions $\chi : \Omega \to \Re^+$ and $\psi : \Omega \to \Re^+$. Denote by $\bar{\chi}$ and $\bar{\psi}$ the expectations of χ and ψ under the probability distribution π, namely $\bar{\chi} = \sum_{\omega\in\Omega} \chi(\omega) \cdot \pi(\omega)$ and $\bar{\psi} = \sum_{\omega\in\Omega} \psi(\omega) \cdot \pi(\omega)$. Then, for every $\alpha < 1$ there exists $\omega^* \in \Omega$ such that*

$$\begin{aligned} \chi(\omega^*) &\leq (1+\alpha)\cdot\bar{\chi} \\ \psi(\omega^*) &\leq \frac{2}{\alpha}\cdot\bar{\psi} \end{aligned}$$

PROOF. From Markov's inequality, we get $\Pr[\chi > (1+\alpha)\cdot\bar{\chi}] < \frac{1}{1+\alpha} \leq 1 - \frac{\alpha}{2}$ and $\Pr[\psi > \frac{2}{\alpha}\cdot\bar{\chi}] < \frac{\alpha}{2}$. Thus $\Pr[\chi \leq (1+\alpha)\cdot\bar{\chi}$ and $\psi \leq \frac{2}{\alpha}\cdot\bar{\psi}] > 0$ as desired. □

The above lemma applied to $\mathcal{P}_i$ with a probability function given by the scaled optimum flow $\mathbf{g}_i/d_i$ for a commodity $i \in \mathcal{I}$, we get the following theorem.

THEOREM 8.2 (ANCHOR THEOREM). *For each commodity i with offline mileage ratio $\rho_i^\Lambda \leq \alpha^2/2$ in an interval $\mathcal{T} = [t_0, t_\ddagger]$, there exists an "anchor" path $\mathbf{B}_i \in \mathcal{P}_i$, for which*

$$\phi'(\mathbf{B}_i, t_0) \leq (1+\alpha)\cdot\frac{\Lambda_i(t_0)}{d_i} \quad (11)$$

$$\nabla^{\phi'}(\mathbf{B}_i, \mathcal{T}) \leq \frac{2}{\alpha}\cdot\rho_i^\Lambda\cdot\frac{\Gamma_i(t_0)}{d_i} \leq \alpha\cdot\frac{\Gamma_i(t_0)}{d_i}. \quad (12)$$

COROLLARY 8.3. *During the interval $\mathcal{T}$, for a non-equilibrium commodity i with offline mileage ratio $\rho_i^\Lambda \leq \alpha^2/2$ and online mileage ratio $\rho_i^\Gamma \leq \alpha$, the cost $\phi'(\mathbf{B}_i)$ of the anchor path can never exceed*

$$\begin{aligned} \phi'(\mathbf{B}_i, t) &\leq \phi'(\mathbf{B}_i, t_0) + \nabla^{\phi'}(\mathbf{B}_i, \mathcal{T}) && (13)\\ &\leq (1+\alpha)\cdot\frac{\Lambda_i(t_0)}{d_i} + \alpha\cdot\frac{\Gamma_i(t_0)}{d_i} && (14)\\ &< \left(\frac{1+\alpha}{1+\tilde{\alpha}}\right)\cdot\frac{\Gamma_i(t_0)}{d_i} + \alpha\cdot\frac{\Gamma_i(t_0)}{d_i} && (15)\\ &= \left(\frac{1+\alpha}{1+5\alpha} + \alpha\right)\cdot\frac{\Gamma_i(t_0)}{d_i} \\ &\leq \left(\frac{1+\alpha}{1+5\alpha} + \alpha\right)\cdot\frac{1}{1-\alpha}\cdot\frac{\Gamma_i(t)}{d_i} && (16)\\ &\leq \frac{1}{1+\alpha}\cdot\frac{\Gamma_i(t)}{d_i}. && (17) \end{aligned}$$

The inequality (13) follows from the definition of $\nabla^{\phi'}$, (14) follows from Theorem 8.2, (15) follows since non-equilibrium commodities satisfy $\Gamma_i(t_0) > (1+\tilde{\alpha})\cdot\Lambda_i(t_0)$, (16) follows from $\Gamma_i(t_0) \leq \Gamma_i(t) + \nabla_i^\Gamma(\mathcal{T})$ since $\nabla_i^\Gamma(\mathcal{T}) \leq \alpha\cdot\Gamma_i(t_0)$, (17) follows from elementary algebra assuming that α is sufficiently small.

COROLLARY 8.4. *For any time t in the interval $\mathcal{T}$, the cost of anchor $\mathbf{B}_i$ for commodity i is at most $\frac{1}{1+\alpha}$ times the average cost for that commodity, and thus the* REROUTE *procedure (Figure 4) attempts to push flow on $\mathbf{B}_i$ at time t.*

We now define parameters τ_0 and $\tau_\dagger$ as follows:

$$\tau_0 = \max_{e\in E} \log_{1+\beta} \frac{2\mu c(e)/\epsilon}{\ddot{\mathbf{f}}(e)} = \tilde{O}\left(\frac{\log m}{\beta}\right) \quad (18)$$

$$\tau_\dagger = \tau_0\cdot 6\cdot H = \tilde{O}\left(\frac{H\log m}{\beta}\right) \quad (19)$$

Note that if an edge e is saturated upward w.r.t. commodity i, i.e., increases its flow $\mathbf{f}_i(e)$ by a factor of $(1+\beta)$ in each of the τ_0 rounds, the flow $\mathbf{f}_i(e)$ exceeds $2\mu c(e)/\epsilon$, which in turn results in the increase of the potential Φ.

THEOREM 8.5. *In any interval of length $\tau_\dagger$ (defined in (19)) or more, each consistently non-equilibrium commodity i contributes a potential reduction of at least*

$$\Delta\Phi \geq \Omega(\alpha\cdot\Gamma_i).$$

PROOF. We proceed by considering the following cases.

Case 1. Suppose that for at least $\tau_\dagger/4$ rounds, we manage to reroute at least $\frac{\beta}{4\cdot H}$ fraction of the demand on paths that are $\Omega(\alpha)$ cheaper than the average. Then, at each such round t, the potential comes down by an amount

$$\Omega\left(\frac{d_i\cdot\beta}{H}\cdot\alpha\cdot\frac{\Gamma_i}{d_i}\right) = \Omega\left(\frac{\alpha\cdot\beta}{H}\cdot\Gamma_i\right). \quad (20)$$

If this is the case for at least $\tau_\dagger/4$ rounds, the total decrease in the potential is at least $\Omega(\alpha\cdot\Gamma_i)$.

Case 2. If on the other hand, the above happens in less than $\tau_\dagger/4$ rounds, we get a contradiction as follows. Note that the downward speed limit is $\frac{\beta}{4\cdot H}$. Thus if we are not able to reroute $\frac{\beta}{4\cdot H}$ fraction of the total demand in each of the $3\tau_\dagger/4$ rounds, it must be the case that the anchor path $\mathbf{B}_i$, which has cost *consistently* and *significantly* smaller than the average cost, is saturated upward. In other words, at least one edge $e \in \mathbf{B}_i$ witnesses an increase of the flow $\mathbf{f}_i(e)$ by a factor of $(1+\beta)$ in each of these rounds. Since there are at most H edges on $\mathbf{B}_i$, it follows that for at least $\frac{3\tau_\dagger}{4\cdot H}$ rounds, some edge $e^* \in \mathbf{B}_i$ has been upward saturated. This leads to a total increase in $\mathbf{f}_i(e^*)$ by a factor of

$$(1+\beta)^{\frac{3\tau_\dagger}{4\cdot H}} > \exp\left(\frac{\tau_\dagger\cdot\beta}{2\cdot H}\right).$$

Now the edge e^* can suffer a reduction in the flow by a factor of at most $1 - \frac{\beta}{4\cdot H}$ due to the downward speed limit in each of the $\tau_\dagger$ rounds. This gives a cumulative reduction by factor of

$$\left(1 - \frac{\beta}{4\cdot H}\right)^{\tau_\dagger} > \exp\left(-\frac{\tau_\dagger\cdot\beta}{3\cdot H}\right).$$

The product of these two terms is at least

$$\exp\left(\frac{\tau_\dagger\cdot\beta}{6\cdot H}\right) > \frac{2\mu c(e)/\epsilon}{\ddot{\mathbf{f}}(e)}.$$

Thus $\mathbf{f}_i(e)$ increases beyond $2\mu c(e)/\epsilon$, contradicting the fact that the overall potential Φ does not ever increase (Corollary 5.3). □

Aggregating non-equilibrium commodities. Theorem 8.5 implies that the overall potential reduction due to consistently non-equilibrium commodities is

$$\Delta\Phi \geq \Omega\left(\sum_{i\in\mathcal{I}} \alpha\cdot\Gamma_i\right) = \Omega(\aleph\cdot\xi^{\mathcal{I}}\cdot\alpha\cdot\Phi).$$

9. CONCLUSIONS

In summary, we have presented the first greedy distributed flow optimization algorithm that converges in the number of rounds that is logarithmic in the number of flow-paths, or in other words, linear in the maximal path length. Since only short paths (less than 10 hops) are used on the Internet, the convergence in time proportional to the path length is a reasonable result in practice. However it is not obvious that better bounds (that do not depend polynomially on the path length) are not possible. This should certainly be a subject of future research. Another issue is whether it is truly necessary to be aggressive while increasing the flow and conservative while decreasing the flow. This may be either an inherent issue or an artifact of our proof technique.

10. REFERENCES

[1] B. Awerbuch and Y. Azar. Local optimization of global objectives: Competitive distributed deadlock resolution and resource allocation. *FOCS*, 1994.

[2] B. Awerbuch, Y. Azar, and A. Epstein. The price of routing unsplittable flow. *STOC*, 2005.

[3] B. Awerbuch, Y. Azar, Y. Richter, and D. Tsur. Tradeoffs in worst-case equilibria. *WAOA*, 2004.

[4] B. Awerbuch and R. Khandekar. Distributed network monitoring and multicommodity flows: a primal-dual approach. *PODC*, 2007.

[5] B. Awerbuch, R. Khandekar, and S. Rao. Distributed algorithms for multicommodity flow problems via approximate steepest descent framework. *SODA*, 2007.

[6] B. Awerbuch and T. Leighton. A simple local-control approximation algorithm for multicommodity flow. *FOCS*, 1993.

[7] B. Awerbuch and T. Leighton. Improved approximation algorithms for the multicommodity flow problem and local competitive routing in dynamic networks. *STOC*, 1994.

[8] B. Awerbuch, B. Patt-Shamir, and G. Varghese. Self-stabilization by local checking and correction. *FOCS*, 1991.

[9] B. Awerbuch and G. Varghese. Distributed program checking: a paradigm for building self-stabilizing distributed protocols. *FOCS*, 1991.

[10] Y. Bartal, J. W. Byers, and D. Raz. Global optimization using local information with applications to flow control. *FOCS*, 1997.

[11] S. Chien and A. Sinclair. Convergence to approximate nash equilibria in congestion games. *SODA*, 2007.

[12] E. Dijkstra. Self stabilizing systems in spite of distributed control. *CACM*, 17:643–644, Nov 1974.

[13] S. Dolev, A. Israeli, and S. Moran. Self-stabilization of dynamic systems assuming only read/write atomicity. *PODC*, 1990.

[14] E. Even-Dar and Y. Mansour. Fast convergence of selfish rerouting. *SODA*, 2005.

[15] S. Fischer, H. Racke, and B. Vocking. Fast convergence to wardrop equilibria by adaptive sampling methods. *STOC*, 2006.

[16] S. Fischer and B. Vocking. Adaptive routing with stale information. *PODC*, 2005.

[17] N. Garg and J. Könemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. *FOCS*, 1998.

[18] N. Garg and N. E. Young. On-line end-to-end congestion control. *FOCS*, 2002.

[19] M. G. Gouda and N. J. Multari. Stabilizing communication protocols. Technical Report TR-90-20, Dept. of Computer Science, University of Texas at Austin, June 1990.

[20] R. M. Karp, E. Koutsoupias, C. H. Papadimitriou, and S. Shenker. Optimization problems in congestion control. *FOCS*, 2000.

[21] A. Khanna and J. A. Zinky. The revised arpanet routing metric. *SIGCOMM*, 1998.

[22] P. Klein, S. Plotkin, C. Stein, and E. Tardos. Faster approximation algorithms for the unit capacity concurrent flow problem with applications to routing and finding sparse cuts. *STOC*, 1990.

[23] P. Klein, S. Plotkin, C. Stein, and E. Tardos. Faster approximation algorithms for the unit capacity concurrent flow problem with applications to routing and finding sparse cuts. *SIAM Journal on Computing*, 1993.

[24] E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. *Lecture Notes in Computer Science*, 1563:404–413, 1999.

[25] J. F. Kurose and K. W. Ross. *Computer Networking, a top down approach featuring the Internet, 3rd ed.* Addison-Wesley Longman, 2004.

[26] C. Labovitz, A. Ahuja, and F. Jahanian. Experimental study of internet stability and wide-area backbone failures. Technical Report CSE-TR-382-98, University of Michigan, 1998.

[27] T. Leighton, F. Makedon, S. Plotkin, C. Stein, E. Tardos, and S. Tragoudas. Fast approximation algorithms for multicommodity flow problem. *STOC*, 1991.

[28] M. Luby and N. Nissan. A parallel approximation algorithm for positive linear programming. *STOC*, 1993.

[29] J. McQuillan, I. Richer, and E. Rosen. The new routing algorithm for the arpanet. *IEEE Trans. on Commun.*, 28(5):711–719, May 1980.

[30] J. M. McQuillan, G. Falk, and I. Richer. A review of the development and performance of the arpanet routing algorithm. *IEEE Trans. on Commun.*, 26(12):1802–1811, Dec. 1978.

[31] V. Mirrokni, M. Goemans, and A. Vetta. Sink equilibria and convergence. *FOCS*, 2005.

[32] S. Plotkin, D. Shmoys, and E. Tardos. Fast approximation algorithms for fractional packing and covering problems. *Math of Oper. Research*, pages 257–301, 1994.

[33] J. Rexford. *Handbook of Optimization in Telecommunications, Chapter on Route optimization in IP networks*. Kluwer Academic Publishers, 2005.

[34] T. Roughgarden. *Selfish Routing*. PhD thesis, Cornell University, Department of Computer Science, 2002.

[35] N. E. Young. Sequential and parallel algorithms for mixed packing and covering. *FOCS*, 2001.