Greedy Distributed Optimization of Multi-Commodity Flows
Draft --- not for distribution

Baruch Awerbuch * Rohit Khandekar †

February 19, 2007

Abstract

While multi-commodity flow is a classical combinatorial optimization problem, it also directly 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 first known stateless distributed optimization algorithms for the concurrent multi-commodity flow problem with poly-logarithmic convergence. More precisely, our algorithm achieves $1+\epsilon$ approximation, with running time $O(\log \mathrm{P} \cdot \log^{O(1)} m \cdot (1/\epsilon)^{O(1)})$ (P is number of paths in the network). No prior results exist for our model.

Viewed from the point of view of classical sequential algorithm, our algorithm is a reasonable alternative to existing polynomial-time sequential approximation algorithms, such as Garg-Könemann [GK98]. The algorithm is rather elegant, (approximately a dozen lines of pseudo-code), and can be be easily implemented or taught in a classroom.

Our result can be contrasted with numerous (thousands) of distributed optimization heuristics for multi-commodity flow published in the course of last four decades, that lack non-trivial provable performance guarantees. 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 existing flow control heuristics are variations of TCP, which uses a conservative cap on the increase (e.g., additive), and rather liberal cap on the decrease (e.g., multiplicative). In contrast, our algorithms requires the increase to be multiplicative, and that this increase is *dramatically more aggressive* than the decrease in the rate. We conclude that rigorous analysis does suggest a drastic change to existing networking building blocks.

*Johns Hopkins University. email: baruch@cs.jhu.edu. Partially supported by NSF grants CCF 0515080, ANIR-0240551, and CCR-0311795, and CNS-0617883.

†IBM T.J. Watson Research Center. email: rkhandekar@gmail.com.

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 that the flows make their routing decisions in cooperative but uncoordinated manner, without having access to global clock and without being able to properly initialize and synchronize their individual executions. The flows will be only observing the current network load, 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 [Dij74, GM90, DIM90, AV91, APV91] which 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 undetermined period of time, and then to wake up. Also, this 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.

- *Asynchronicity:* Algorithms should not be driven by global clock, and should work in case when local clocks have bounded drifts.

Existing synchronous/asynchronous path-based flow algorithms [AA94, LN93, BBR97, You01, AKR07] don't work in stateless model; their performance bounds crucially depend on maintaining a state (e.g., proper initialization).

Our objective is to replace for 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 describe now these building blocks.

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 load sensitive, and respond dynamically to changing congestion patterns.

Originally, Internet traffic has been congested-driven, and different flows tried to re-route themselves along least congested paths [MFR78]. The following phenomenon, referred as "tragedy of the commons" has been detected by the practitioners [MRR80]. 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. Thus, the "greedy" policy of routing on cheapest/least congested path, which is the goal of "traffic engineering", leads to unpredictable oscillations and wastes network resources.

In order to avoid instability problem, Internet routing protocols since the seventies [MRR80] avoid dynamic (congestion caused) re-routing altogether [LAJ98, KZ98, Rex05, KR04], and thus operate over

static and potentially significantly sub-optimal paths. Since current Internet routing is load-insensitive minimum-hop routing, it is obviously more vulnerable to attacks than adaptive load-adjusted routing proposed in this work. One of the consequences is that current Internet routing architecture is not supporting quality of service guarantees and most basic security guarantees. Uptill now, the inability to deal with congestion in an adaptive manner by re-routing 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 ceratin 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, then the flow does not need to observe the above downward speed limit. Flow control is *necessary* since otherwise concurrent operation with greedy users certainly leads to collapse.

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. A presents our definition and proof of near-optimality of aggregate equilibria. Sec. B (Appendix) analyzes flow dynamics; algorithm's parameters are explained in Sec. 6, followed by a (non-trivial) proof of fast convergence to equilibrium.

2 The statement of the problem

Billboard computing model. We use the following simplified model of distributed computation. Agents associated with different commodities (flows) are making changes to routing choices for their flows. Each edge is assumed to maintain a billboard that records its current congestion. In each round of a global clock, the agents concurrently "read" the congestion values at different edges from the billboards, and subsequently concurrently "write" their changes to these billboards; these changes will be observable in the next round. Each agent is assumed to be oblivious to the existence of other agents and is not allowed to communicate explicitly with other agents.

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 end-point (source) of the flow the database of all the network links, *including* current utilization of that link. Maintaining local link state databases at the flow sources means that utilization of all the links must be continuously reflooded through the network, resulting in larger (but still acceptable) maintenance cost.

Our algorithm, in fact, works *as is* in a partially asynchronous model, without synchronized rounds assumption, i.e., local clocks may have different drifts and phases; this causes only a minor complication to the proofs. 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 [PST94, GK98] or by statefull distributed algorithm [AKR07] that requires initialization. In fact it is not even a-priori obvious that near-optimal distributed stateless solutions even *exist*.

Formal Statement of Multi-commodity Flow problem. Informally, the above network bandwidth management problems can be modeled as multi-commodity flow problems in a directed capacitated graph, with a collection of *commodities*, each characterized by the following: source (where the flow is originated), sink (where the flow ends up), and demand (the amount of flow available). The normalized load of each edge is the ratio between flow on this edges and its capacity. The flow for each commodity must satisfy

Kirkhoff laws of flow conservation at intermediate points (except for source and destination). Maximum concurrent flow MCF maximizes the minimal ratio between the flow of a commodity and its demand, or equivalently, minimizes maximum edge load while meeting all flow demands. This is exactly the *pure routing* problem in networks.

More precisely, consider a directed capacitated graph (V, E, c) where $c : E \to R^+$ is the capacity assignment to edges, and a set of commodities $\mathcal{U} \subset V \times V$, with a source s_i and a sink t_i for each commodity. We have demand $d_i > 0$ for each such commodity $i \in \mathcal{U}$. Our goal is to find a collection of flow assignments for each commodity i, $\mathbf{f} = \{\mathbf{f_i}\}$ where $\mathbf{f} : \mathcal{U} \times E \to R^+$, such that aggregate flow is satisfying the flow and capacity constraints, while minimizing the maximum edge load:

$$|\mathbf{f}| = \max_{e \in E} \ell(e) \quad \text{where} \quad \ell(e) = \frac{f(e)}{c(e)} = \frac{\sum_{i \in \mathcal{U}} f_i(e)}{c(e)}. \tag{1}$$

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 local computational overhead imposed by the local program of each agent, in terms of the number of local computational steps.

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

Notation 2.1 Let m, n, k be number of edges, vertices, and flow commodities in the network graph; 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 paths, respectively. A flow algorithm has *poly-logarithmic convergence* if its convergence time is $O(\log \mathrm{P} \log^{O(1)} m)$.

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 loaded servers. This corresponds to path length $H = 1$, i.e., routing in bipartite graph. In summary, convergence in time proportional to path length is a reasonable result in practice.

3 Our results and techniques

We provide *first known* stateless distributed optimization algorithms for the concurrent multi-commodity flow problem with *poly-logarithmic convergence*. Corollary C.7 (Sec. C.2) and Thm. C.2 yield

Theorem 3.1 Our algorithm (see Sec. 5, in Figures 2,3, 4, 5) achieves $1 + \epsilon$ approximation, with convergence time $O(\log \mathrm{P} \cdot \log^{O(1)} m \cdot (1/\epsilon)^{O(1)})$.

The *computational overhead* is comparable to that of blocking flow; it can be easily upper-bounded by $\tilde{O}(H \cdot m^2)$ (per commodity). Viewed from the point of view of classical sequential algorithms, our algorithm is only slightly inferior to $\tilde{O}(m^2)$ time best existing sequential approximations [GK98]. The algorithm is rather elegant (approximately a dozen lines of pseudo-code) and is easily implementable.

No prior results exist in stateless model (Sec. 4). Our result can be contrasted with numerous (thousands) of distributed optimization heuristics for multi-commodity flow published in the course of a

number of decades, that lack non-trivial provable performance guarantees. 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 existing flow control heuristics are variations of TCP, which uses a conservative cap on the increase (e.g., additive), and 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 in the rate. Thus, our rigorous analysis does suggest a drastic change to existing networking building blocks.

We stress that our algorithm reaches near-optimality *without* ever reaching even approximate Nash equilibrium in the underlying metric. In fact, we prove that for every convex function, near-optimality is implied by a *much weaker* notion of *aggregate equilibrium*, which only requires that *average* local imbalance is small, with average weighed by relative contributions of flows to potential.

The proof of our algorithm (see appendix) is very involved; the proof technique is completely unrelated to techniques in recent work on distributed primal dual algorithms such as [AKR07, AK]. The crux of our proof is the fact that potential reduction over long enough time intervals is proportional to average local imbalance. Thus, while we are far from optimum, constant fraction of potential is improved.

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 (sec. A.2). 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 (inversely proportional to number of commodities) fraction of the demand of each commodity, i.e. using tiny *additive* speed limit on the way we change our flow. However, this results in convergence time which is linear in the number of commodities, similarly to sequential flow algorithms such as [KPST91, KPST93, LMP$^+$91, PST91]. The latter work proves convergence under assumption that flows proceed *sequentially* routing small fraction of their demand along short paths. These proofs are not useful for us since they require commodities to be serialized and since this results in very large convergence time; our goal is to achieve convergence times which are logarithmic in number of network paths or linear in path length (recall that path length is rather short in practice).

It is essentially impossible to achieve this type of convergence without allowing constant fraction of the demand being 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 distributed primal-dual algorithms such as [AKR07] succeed in accomplishing this "pumping up" effect *using 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* and preventing us from using techniques of [AKR07, GK98]. In [FRV06] a proof of concurrent potential reduction in non-monotonic case is accomplished for a special case of a single commodity. Handling multiple commodities (as in this paper) is 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 the 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 paths corresponding to cheap paths used by optimum solution get saturated upwards, or large fraction of the potential gets reduced. Since the downward speed limit are less aggressive, the "downwards" drift cannot counter-act the "upwards" drift, 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 only correct if the optimum cheap path is consistently cheaper than the cost on the flow paths chosen

Reference	Model restrictions	Convergence	Computation
[LN93, You01]	requires state	$\tilde{O}(\log \mathrm{P}) = \tilde{O}(H)$	$\tilde{O}(\mathrm{P}) = \tilde{O}\left(n^H\right)$
[FRV06]	single commodity	$\tilde{O}\left(\log^{O(1)} \mathrm{P}\right) = \tilde{O}\left(H^{O(1)}\right)$	$\tilde{O}(\mathrm{P}) = \tilde{O}\left(n^H\right)$
[AKR07]	requires state	$\tilde{O}(\log \mathrm{P}) = \tilde{O}(H)$	$\tilde{O}\left(\log \mathrm{P} \cdot m^2\right) = \tilde{O}\left(H \cdot m^2\right)$
[AK]	requires state	$\tilde{O}\left(\log^3 \mathrm{P}\right) = \tilde{O}\left(H^3\right)$	$\tilde{O}\left(\log^3 \mathrm{P} \cdot m \cdot k\right) = \tilde{O}\left(H^3 \cdot m \cdot k\right)$
This paper	no restrictions	$\tilde{O}(\log \mathrm{P}) = \tilde{O}(H)$	$\tilde{O}\left(\log \mathrm{P} \cdot m^2\right) = \tilde{O}\left(H \cdot m^2\right)$

Figure 1: Our results vs. existing work. H: path length; P: # paths, m: #edges, n: #nodes. $\tilde{O}(\cdot)$ absorbs polylog factors.

by the algorithm. To deal with fluctuations, we show that if cost of the dynamic flow paths used by a commodity fluctuates, or if all costs of all the cheap paths used by optimum fluctuate, then large fraction of the discrepancy between optimum and online costs is being "burnt" (Sec. B.6).

In the analysis of the equilibrium (Sec. A) 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 if due to out-of-equilibrium commodities, then above argument holds about potential coming downby a constant factor in logarithmic number of steps.

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 (Sec. C.2). This is a significant deviation from the previous primal-dual exponential-potential-based algorithms [GK98, You01, AKR07]. 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. [PST91].

4 Existing results

The comparison of our result with existing work is summarized in Fig. 1 and 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 ground-breaking results by Even-Dar et al. [EDM05] and Fisher et al. [FV05, FRV06] who state results comparable to our paper in case that commodities operate in a *complete* network (clique). It appears that [FRV06] 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 [FV05, FRV06] is that its computational overhead is inherently *exponential* (a distinct "agent" needs to be assigned to each path). An open problem stated in [FRV06] is to reduce the computational overhead to polynomial and generalizing to multiple commodities. It is worth to point out that [FV05, FRV06] 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 [KP99, Rou02, AAE05, AART04, CS06, MGV05] 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 re-routing along shorter paths decreases the natural potential function, thus causing (eventual) convergence to (approximate) Nash equilibrium, potentially after a long time. Recent work [CS06, MGV05] presents *polynomial* bounds on convergence to equilibria. Essentially all of this work uses *polynomial* objective functions; while in order to minimize network load one needs to use *exponential* cost function, as in our paper. Note that for our purpose of reaching optimum, approximate equilibrium notion used in [CS06, MGV05] 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. [FV05, FRV06], this research direction fails to 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 re-routings 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 achieve our goal of designing 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 [GY02a, LN93, BBR97, You01, AKR07, AK]. The initial work in this area [GY02a, LN93, BBR97, You01] 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 [AKR07], who provided first statefull algorithm with *polynomial* computational overhead and poly-log distributed convergence. Some of these algorithms, e.g., [AKR07, AK] are *poly-logarithmically more efficient* than our algorithm. It is is crucial to note that these statefull solutions *fail* in our stateless model since these algorithms need to be *consistently initialized*.

Flow control. There is also a fair amount of research on "reverse-engineering" the Internet, namely understanding what kind of optimization problems are solved by existing (statefull) flow control mechanisms and showing that under certain assumptions some interesting properties can be proved about TCP flow control [KKPS00] or [GY02b]; such results are incomparable to our work.

Stateless algorithms for Routers model. In Billboard model used in the current paper we have dumb routers and all the intelligence at the end-to-end users (flow agents). In the Routers model the intelligence is at the routers, with dumb end-to-end users (flow agents). Routers model avoids using billboards and link state databases by leveraging the abilities of network routers. It does not assume that individual routers are aware of the topology. It only assumes that routers are aware of different flow commodities. For each commodity, routers need to pick an adjacent edge over which the flow of that commodity will be forwarded. The routing algorithms for this model are also called "distance vector" algorithms, because they maintain a "proximity" metric for each destination, with packets flowing the direction of decreased metric. An example of stateless multi-commodity flow algorithm in the Routers model is multi-commodity flow result by Awerbuch and Leighton [AL93, AL94]. This algorithm cannot be used in our model because the actual flows sent in this algorithm violate flow preservation at intermediate nodes, accumulating flow excess at nodal queues. In our Billboard model, flows must obey conservation laws; thus these results are not comparable.

5 Description of the algorithm

5.1 Greedy Concurrent Re-routing Framework

The purpose this section is to formally capture 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*, *flow control*, and *greedy re-routing*. The detailed description of our algorithm is presented in Sec. 5.2.

Routing metric: is the exponential function indicating how "cost" of an edge grows with its congestion. This metric is imposed to encourage the agents to re-route their flows towards less congested paths.

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

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

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

5.2 Detailed description of the algorithm

Notations: We denote the inertia threshold (min. profit margin) by α. ϵ is the approximation guarantee. $f(e)$ is flow over edge e and $c(e)$ is capacity of e; ℓ is the maximal load observed in the network. $f_i(e)$ is flow of commodity i on e. $\phi'_\mu(f(e))$ is the cost of edge e used in the routing metric. m, n are number of edges and vertices. $\mathcal{P}_i$ and s_i are the set of paths and the source for commodity i. H is maximum number of edges in any path considered in our solution. Upward (downward) residual capacities are denoted by $\Delta_i^+(e)$ ($\Delta_i^-(e)$) respectively.

Main program: is defined in Proc. MAIN (see Fig. 2). Upon the beginning of each round, it reads the flows values from billboard, and executes procedures ROUTEMETRIC , FLOWCONTROL and GREEDYREROUTE. Subsequently, the final value of flow for commodity i is finally written on billboard.

1. **Input (read)** flow vector $\mathbf{f_i}$
2. **Call Procedures** ROUTEMETRIC , FLOWCONTROL, GREEDYREROUTE
3. **Output (write)** flow vector $\mathbf{f_i}$

Figure 2: Procedure MAIN for commodity i.

Procedure RouteMetric for commodity i

1. **if** $\mu < \dfrac{|\mathbf{f}| \cdot \epsilon}{2}$ **or** $\mu > |\mathbf{f}| \cdot \epsilon$, **then** set $\mu \leftarrow \dfrac{|\mathbf{f}| \cdot \epsilon}{2}$
2. $\forall e \in E$, set $\phi_{e,\mu}(f(e)) \equiv m^{\frac{f(e)}{c(e)\cdot\mu}}$ and compute $\phi'_{e,\mu}(f(e))$
3. $\forall \mathbf{A} \in \mathcal{P}_i$, define $\phi'_\mu(\mathbf{A}) \equiv \sum_{e \in \mathbf{A}} \phi'_{e,\mu}(f(e))$

Figure 3: RouteMetric procedure sets μ and edge/path costs.

Procedure FlowControl for commodity i

1. $\alpha \leftarrow \frac{\epsilon}{4 \log m}$
2. $\beta \leftarrow \frac{\alpha}{8} \cdot \frac{\epsilon}{\log m}$
3. $\ddot{f}_i(e) \leftarrow \frac{\mu}{\log_2 m} \cdot \frac{c(e)}{k} \cdot \log_2(1 + \frac{\alpha}{8})$
4. **upon each round**, $\forall e \in E$, set $\tilde{f}_i(e) \leftarrow f_i(e)$ /* `remember flow` */
5. **maintain** $\forall e \in E,\ \Delta_i^-(e) \leftarrow f_i(e) - \tilde{f}_i(e)(1 - \dfrac{\beta}{4 \cdot H})$ /* `downward (pull) capacity`*/
6. **maintain** $\forall e \in E,\ \Delta_i^+(e) \leftarrow (1+\beta) \cdot \max\left\{\tilde{f}_i(e), \ddot{f}(e)\right\} - f_i(e)$ /* `upward (push) capacity`*/

Figure 4: Procedure FlowControl for commodity i. m is #edges.

Procedure GreedyReroute for commodity i

1. **while** $\exists \mathbf{A} \in \mathcal{P}_i \mid \min_{e \in \mathbf{A}} \Delta_i^+(e) > 0\ ,\ d_i \cdot (1+\alpha) \cdot \phi'_\mu(\mathbf{A}) < \sum_e f_i(e)\phi'(e)$ **and** $\sum_{v \in V} \Delta_i^-(s_i, v) > 0$
   (a) $\mathbf{A} \leftarrow$ argmin $\phi'_\mu(\mathbf{A}')$ over non-empty set of $\mathbf{A}'$ meeting condition in line 1.
   (b) $\delta \leftarrow \min\left\{ \min_{e \in \mathbf{A}} \Delta_i^+(e)\ ,\ \sum_{v \in V} \Delta_i^-(s_i, v) \right\}$ `/* min of push & pull amounts */`
   (c) $\forall e \in E,\ f_i(e) \leftarrow f_i(e) - \delta \cdot \dfrac{f_i(e)}{d_i}$ and $\Delta_i^-(e) \leftarrow \Delta_i^-(e) - \delta \cdot \dfrac{f_i(e)}{d_i}$
   (d) $\forall e \in \mathbf{A},\ f_i(e) \leftarrow f_i(e) + \delta$ and $\Delta_i^+(e) \leftarrow \Delta_i^+(e) - \delta$

Figure 5: Procedure GreedyReroute for commodity i.

Routing metric: is defined in Proc. RouteMetric (Fig. 3). Parameter μ is continuously decreased in proportion to maximum load $|\mathbf{f}|$. 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 $\phi_{e,\mu}(f(e))$.

Residual capacity and inertia threshold Upward ("push") and downward ("pull") residual capacities $\Delta_i^+(e)$ and $\Delta_i^-(e)$ (and parameter β) are assigned in Proc. FlowControl in Fig. 4. Here $\tilde{f}_i(e)$ denotes the flow of commodity i on edge e in the beginning of the step. Note the crucial multiplicative gap proportional to max path length H between downwards and upwards residual capacities. A flow can re-route only if the process of re-routing improves its routing metric cost at least by factor of $1+\alpha$.

Greedy augmentation: is done by Proc. GreedyReroute in Fig. 5. It *pushes* flow forward over cheapest path, 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 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 cheap path. The allowed pull-backward amount is sum of downward capacities of source's outgoing edges. If the profit margin is met, then the minimum of the pull and push amounts is allowed to circulate, decreasing upward (downward) capacities appropriately.

6 The reasons for specific parameters chosen

We now explain parameters used in the algorithm. Let $\pi(\ell(e)) = m^{\frac{\ell(e)}{\mu}}$.

$$\vartheta_1 = \min_{0\le\ell\le|\mathbf{f}|} \frac{\pi(\ell)}{\pi'(\ell)\cdot\ell} \quad , \quad \text{and} \quad \vartheta_2 = \min_{0\le x\le|\mathbf{f}|} \frac{\pi'(\ell)}{\pi''(\ell)\cdot\ell} \quad \text{and} \quad \vartheta_3 = \min_{0\le\ell\le|\mathbf{f}|} \frac{\pi'(\ell)}{\pi''(\ell)\cdot|\mathbf{f}|} \tag{2}$$

Using Corollary C.4, for exponential function, we have

$$\vartheta_1, \vartheta_2, \vartheta_3 \;\ge\; \frac{\mu}{|\mathbf{f}|\cdot\log m} \ge \frac{\epsilon}{(1+\epsilon)\cdot 2\cdot\log m} = \Omega(\frac{\epsilon}{2\cdot\log m}) \tag{3}$$

Choice of potential approximation $\aleph_\dagger$. We choose $\aleph_\dagger < 1$ as the desired approximation of the potential compared to potential of optimum flow. For our purposes, it is sufficient to choose $\aleph_\dagger = 1/2$.

Choice of approximation threshold α. We choose:

$$\alpha \;=\; \frac{\aleph_\dagger\cdot\vartheta_1}{4} \tag{4}$$

Choice of multiplicative increase β. We are going to define the speed limit β as a parameter such that, for the worst possible online load values, $0 \le \ell \le |\mathbf{f}|$, the *multiplicative* increase/decrease by β fraction in the value of the total load yields only $\alpha/8$ fraction increase/decrease in derivative of the potential ϕ'. Namely, the speed limit is set in such a way that a derivative of the potential can increase/decrease its flow at most by a fraction of $\alpha/8$ (i.e. factor of $1+\alpha/8$).

$$\pi'(\ell+\ell\beta) \;\approx\; \pi'(\ell) + \pi''(\ell)\cdot\ell\cdot\beta < \pi'(\ell) + \pi'(\ell)\cdot\frac{\alpha}{8} \quad \text{or, in other words,} \tag{5}$$

$$\beta \;\le\; \frac{\alpha}{8}\cdot\frac{\pi'(\ell)}{\ell\cdot\pi''(\ell)} = \frac{\alpha}{8}\cdot\frac{\pi'(\ell)}{\pi''(\ell)\cdot\ell} < \frac{\alpha}{8}\cdot\vartheta_2 = \Theta(\alpha\cdot\vartheta_2) \tag{6}$$

Choice of additive speed limit $\ddot{f}_i(e)$. Note that if flow of a certain commodity over an edge is small (e.g., zero) the multiplicative increase in ineffective. To bootstrap increase in the flow, we need to pretend that there is exists a virtual lower bound (offset) on flow of each commodity on each edge. The interpretation of offset $\ddot{f}_i(e)$ is that increase of $\beta \cdot \ddot{f}_i(e)$ for commodity i is allowed regardless of value of $f_i(e)$. We choose $\ddot{f}_i(e)$ so that, the *additive* increase/decrease yields only $\alpha/8$ fraction increase/decrease in derivative of the potential ϕ', i.e.,

$$\ddot{f}_i(e) \quad = \quad \frac{\mu}{\log_2 m} \cdot \frac{c(e)}{k} \cdot \log_2(1 + \frac{\alpha}{8}) \tag{7}$$

All together k commodities contribute additive increase of $k \cdot \ddot{f}(e)$ at the most, load of at most

$$k \cdot \ddot{f}(e) \quad \leq \quad c(e) \cdot \frac{\mu}{\log_2 m} \cdot \log_2(1 + \frac{\alpha}{8}) \tag{8}$$

translating into multiplicative increase in cost of

$$m^{\frac{1}{c(e)\cdot\mu} \cdot \frac{c(e)\cdot\mu}{\log_2 m} \cdot \log_2(1+\frac{\alpha}{8})} = 1 + \frac{\alpha}{8} \tag{9}$$

Summary, conclusions and open problems

In summary, we have presented and proved correctness and running time bounds for the first greedy distributed flow optimization algorithms that convergence in logarithmic time in number of network paths, or in other words, linear in maximal path length. In practice, only short paths (less than 10 hops) are used on the Internet, so convergence in time proportional in path length is a reasonable result in practice. While this result is superior to all existing work in terms of distributed running time, it is not obvious that better bounds (that do not depend polynomially on path length) are not possible. This should certainly be subject of future research.

References

[AA94] Baruch Awerbuch and Yossi Azar. Local optimization of global objectives: Competitive distributed deadlock resolution and resource allocation. In *Proc. 35th IEEE Symp. on Found. of Comp. Science*, 1994.

[AAE05] B. Awerbuch, Y. Azar, and A. Epstein. The price of routing unsplittable flow. In *Proceedings of ACM STOC*, 2005.

[AART04] Baruch Awerbuch, Yossi Azar, Yossi Richter, and Dekel Tsur. Tradeoffs in worst-case equilibria. In *Proceedings of WAOA*, 2004.

[AK] Baruch Awerbuch and Rohit Khandekar. Distributed network monitoring and multicommodity flows:primal-dual approach. Submitted for publication.

[AKR07] Baruch Awerbuch, Rohit Khandekar, and Satish Rao. Distributed algorithms for multicommodity flow problems via approximate steepest descent framework. In *In Proc. ACM SIAM Conf. on Applied Algs. (SODA)*, 2007.

[AL93] Baruch Awerbuch and Tom Leighton. A simple local-control approximation algorithm for multicommodity flow. In *Proc. 34th IEEE Symp. on Found. of Comp. Science*, pages 459–46. IEEE, November 1993.

[AL94] Baruch Awerbuch and Tom Leighton. Improved approximation algorithms for the multicommodity flow problem and local competitive routing in dynamic networks. In *Proc. 26th ACM Symp. on Theory of Computing*, May 1994.

[APV91] Baruch Awerbuch, Boaz Patt-Shamir, and George Varghese. Self-stabilization by local checking and correction. In *Proc. 32nd IEEE Symp. on Found. of Comp. Science*, pages 268–277, October 1991.

[AV91] Baruch Awerbuch and George Varghese. Distributed program checking: a paradigm for building self-stabilizing distributed protocols. In *Proc. 32nd IEEE Symp. on Found. of Comp. Science*, October 1991.

[BBR97] Yair Bartal, John W. Byers, and Danny Raz. Global optimization using local information with applications to flow control. In *Proc. 38th IEEE Symp. on Found. of Comp. Science*, pages 303–312, 1997.

[CS06] Steve Chien and Alistair Sinclair. Convergence to approximate nash equilibria in congestion games, 2006.

[Dij74] E.W. Dijkstra. Self stabilizing systems in spite of distributed control. *CACM*, 17:643–644, Nov 1974.

[DIM90] Shlomo Dolev, Amos Israeli, and Shlomo Moran. Self-stabilization of dynamic systems assuming only read/write atomicity. In *Proc. 10th ACM Symp. on Principles of Distrib.Computing*, Quebec City, Canada, August 1990.

[EDM05] E. Even-Dar and Y. Mansour. Fast convergence of selfish rerouting. pages 772–781, 2005.

[FRV06] Simon Fischer, Haralde Racke, and Berthold Vocking. Fast convergence to wardrop equilibria by adaptive sampling methods. In *ACM STOC (Symposium on Theeoretical Computer Science)*, 2006.

[FV05] S. Fischer and B. Vocking. Adaptive routing with stale information. In *In Proc. 24th Ann. ACM SIGACT-SIGOPS Symp. on Principles of Distributed Computing (PODC), pages 276–283, 2005.*, 2005.

[GK98] N. Garg and J. Könemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, pages 300 – 309, Palo Alto, CA, November 1998.

[GM90] Mohamed G. Gouda and Nicholas J. Multari. Stabilizing communication protocols. Technical Report TR-90-20, Dept. of Computer Science, University of Texas at Austin, June 1990.

[GY02a] N. Garg and N. E. Young. On-line end-to-end congestion control. In *IEEE Symposium on Foundations of Computer Science*, pages 538–546, 2002.

[GY02b] Naveen Garg and Neal E. Young. On-line, end-to-end congestion control. In *Proceedings of IEEE Symposium on Foundations of Computer Science*, 2002.

[KKPS00] Richard M. Karp, Elias Koutsoupias, Christos H. Papadimitriou, and Scott Shenker. Optimization problems in congestion control. In *IEEE Symposium on Foundations of Computer Science*, pages 66–74, 2000.

[KP99] Elias Koutsoupias and Christos Papadimitriou. Worst-case equilibria. *Lecture Notes in Computer Science*, 1563:404–413, 1999.

[KPST91] P. Klein, S.A. Plotkin, C. Stein, and E. Tardos. Faster approximation algorithms for the unit capacity concurrent flow problem with applications to routing and finding sparse cuts. Technical Report 961, School of Operations Research and Industrial Engineering, Cornell University, 1991. A preliminary version of this paper appeared in stoc22, 1990, pages 310-321.

[KPST93] 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. To appear.

[KR04] James F. Kurose and Keith W. Ross. *Computer Networking, a top down approach featuring the Internet, 3rd ed.* Addison-Wesley Longman, 2004.

[KZ98] Atul Khanna and John A. Zinky. The revised arpanet routing metric. In *Proc. ACM SIGCOMM*, page 4556, Sept. 1998.

[LAJ98] 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.

[LMP+91] T. Leighton, F. Makedon, S. Plotkin, C. Stein, E. Tardos, and S. Tragoudas. Fast approximation algorithms for multicommodity flow problem. In *Proc. 23rd ACM Symp. on Theory of Computing*, pages 101–111, May 1991.

[LN93] Michael Luby and Noam Nissan. A parallel approximation algorithm for positive linear programming. In *Proc. 25th ACM Symp. on Theory of Computing*, pages 448–457, May 1993.

[MFR78] J. M. McQuillan, G. Falk, and I. Richer. A review of the development and performance of the arpanet routing algorithm. *IEEE Trans. on Communications*, COM-26, No. 12:1802–1811, December 1978.

[MGV05] V. Mirrokni, M. Goemans, and A. Vetta. Sink equilibria and convergence. In *Proceedings of IEEE Symposium on Foundations of Computer Science*, 2005.

[MRR80] John McQuillan, Ira Richer, and Eric Rosen. The new routing algorithm for the arpanet. *IEEE Trans. on Commun.*, 28(5):711–719, May 1980.

[PST91] S. Plotkin, D. Shmoys, and E. Tardos. Fast approximation algorithms for fractional packing and covering problems. In *Proc. 32nd IEEE Symp. on Found. of Comp. Science*, 1991.

[PST94] S. Plotkin, D. Shmoys, and E. Tardos. Fast approximation algorithms for fractional packing and covering problems. *Math of Oper. Research*, pages 257–301, 1994.

[Rex05] Jennifer Rexford. *Handbook of Optimization in Telecommunications, chapter on Route optimization in IP networks.* Kluwer Academic Publishers, 2005.

[Rou02] T. Roughgarden. *Selfish Routing.* PhD thesis, Cornell University, Department of Computer Science, 2002. See also `http://www.cs.cornell.edu/timr/`.

[You01] Neal E. Young. Sequential and parallel algorithms for mixed packing and covering. In *IEEE Symposium on Foundations of Computer Science*, pages 538–546, 2001.

Appendix: Proofs

A Defining Aggregate Equilibria

A.1 Auxiliary Potentials Definitions

In the notation below, we denote $\mathbf{f}$ the current "online" flow in our algorithm, and $\mathbf{g}$ the optimum "offline" flow. We define now *online* potential $\gamma(e)$ and offline potential $\lambda(e)$ defined for each edge; we also define and corresponding shares for commodity i of such potentials, $\gamma_i(e)$ and $\lambda_i(e)$:

$$\lambda(e) = g(e) \cdot \phi'_{e,\mu}(f(e)) \quad \text{and} \quad \lambda_i(e) = \lambda(e)\frac{g_i(e)}{g(e)} = \phi'_{e,\mu}(f(e)) \cdot g_i(e) \tag{10}$$

$$\gamma(e) = f(e) \cdot \phi'_{e,\mu}(f(e)) \quad \text{and} \quad \gamma_i(e) = \gamma(e)\frac{f_i(e)}{\sum_j f_j(e)} = \phi'_{e,\mu}(f(e)) \cdot f_i(e) \tag{11}$$

and we define the total contribution of all edges to commodity as

$$\Gamma_i = \sum_e f_i(e) \cdot \phi'_{e,\mu}(f(e)) \quad \text{and} \quad \Gamma = \sum_e \gamma_i(e) = \sum_i \Gamma_i \tag{12}$$

$$\Lambda_i = \sum_e g_i(e) \cdot \phi'_{e,\mu}(f(e)) \quad \text{and} \quad \Lambda = \sum_e \lambda_i(e) = \sum_i \Lambda_i \tag{13}$$

A.2 Convexity argument

Consider arbitrary solution $\mathbf{f}$ and suppose $\mathbf{g} =$ argmin ℓ. Consider any convex (e.g., exponential) function $\phi(\mathbf{f})$. Consider a convex combination point $\mathbf{z} = (1-\eta)\cdot\mathbf{f} + \eta\cdot\mathbf{g}$. By convexity of ϕ, value at a convex combination $\phi(\mathbf{z})$ is below convex combination of values $(1-\eta)\cdot\phi(\mathbf{f}) + \eta\cdot\phi(\mathbf{g})$, namely

$$(1-\eta)\cdot\phi(\mathbf{f}) + \eta\cdot\phi(\mathbf{g}) > \phi\Big((1-\eta)\cdot\mathbf{f} + \eta\cdot\mathbf{g}\Big) \tag{14}$$

Now, for small η, using linear approximation for Taylor series, re-arranging & dividing by η yields:

$$\phi\Big((1-\eta)\cdot\mathbf{f} + \eta\cdot\mathbf{g}\Big) \approx \phi(\mathbf{f}) - \eta\cdot(\phi'(\mathbf{f})\cdot\mathbf{f} - \phi'(\mathbf{f})\cdot\mathbf{g}) \tag{15}$$

$$\phi(\mathbf{f}) - (1-\eta)\cdot\phi(\mathbf{f}) - \eta\cdot\phi(\mathbf{g}) < \eta\cdot(\phi'(\mathbf{f})\cdot\mathbf{f} - \phi'(\mathbf{f})\cdot\mathbf{g}) \tag{16}$$

$$\phi(\mathbf{f}) - \phi(\mathbf{g}) < \phi'(\mathbf{f})\cdot\mathbf{f} - \phi'(\mathbf{f})\cdot\mathbf{g} = \boldsymbol{\Gamma} - \boldsymbol{\Lambda} \tag{17}$$

A.3 Aggregate equilibrium definition

Definition A.1 We say that the flows are in *aggregate ξ-equilibrium* w.r.t. a fixed parameter $\tilde{\alpha} = 4\cdot\alpha$ and a time interval $\mathcal{T} = [t_0, t_\ddagger]$ if there exists a partition of commodities $\mathcal{U} = \mathcal{C}_\mathcal{T} \bigcup \mathcal{O}_\mathcal{T}$, into equilibrium and non-equilibrium commodities $\mathcal{C}_\mathcal{T}, \mathcal{O}_\mathcal{T}$ so that:

- *equilibrium commodities $\mathcal{C}_\mathcal{T}$:* those for whom Λ_i is *always* not much worse than Γ_i during $\mathcal{T}$;

$$\mathcal{C}_\mathcal{T} = \{i \mid \forall t \in \mathcal{T}, \Gamma_i(t) \le (1+\tilde{\alpha})\cdot\Lambda_i(t)\} \tag{18}$$

- *non-equilibrium commodities $\mathcal{O}_{\mathcal{T}}$:* those for whom Γ_i is *occasionally* much worse than Λ_i, and $\xi_{\mathcal{T}}$, the relative contribution to $\Gamma(t_0)$ of these types of commodities at t_0 is less than ξ:

$$\mathcal{O}_{\mathcal{T}} = \{i \mid \exists t \in \mathcal{T}, \Gamma_i(t) > (1+\tilde{\alpha}) \cdot \Lambda_i(t)\} \; ; \quad \text{and} \quad \xi_{\mathcal{T}} = \frac{1}{\Gamma(t_0)} \cdot \sum_{i \in \mathcal{C}_{\mathcal{T}}} \Gamma_i(t_0) < \xi \tag{19}$$

Lemma A.1 For each interval of time $\mathcal{T}$, one the following properties holds at its initial time t_0:

$$\aleph = \frac{\Phi - \Phi^*}{\Phi} < \frac{\aleph_\dagger}{2} \qquad \text{i.e., we have achieved the desired approximation threshold} \tag{20}$$

$$\xi > \frac{\vartheta_1 \cdot \aleph_\dagger}{4} \qquad \text{holds if the flows are in aggregate } \xi\text{-equilibrium} \tag{21}$$

Proof. For beginning t_0 of interval $\mathcal{T}$ (indices t_0 and $\mathcal{T}$ are omitted in the equations below), holds:

$$\Gamma - \Lambda \leq \sum_{i \in \mathcal{U}} \phi'(\mathbf{f}) \cdot (\mathbf{f_i} - \mathbf{g}_i) = \sum_{k \in \mathcal{C}} \phi'(\mathbf{f}) \cdot (\mathbf{f_k} - \mathbf{g}_k) + \sum_{j \in \mathcal{O}} \phi'(\mathbf{f}) \cdot (\mathbf{f_j} - \mathbf{g}_j) \tag{22}$$

$$< \sum_{k \in \mathcal{C}} \phi'(\mathbf{f}) \cdot \mathbf{f_k} \cdot (1 - \frac{1}{1+\tilde{\alpha}}) + \sum_{j \in \mathcal{O}} \phi'(\mathbf{f}) \cdot \mathbf{f_j} < \Gamma \cdot (\xi + \frac{\tilde{\alpha}}{1+\tilde{\alpha}}) \approx \Gamma \cdot (\xi + \tilde{\alpha}) \tag{23}$$

Recalling Eqn. 2 defining parameter ϑ_1, we can derive the following fact:

$$\Phi(\mathbf{f}) = \sum_e \phi(f(e)) = \sum_e \pi(\ell(e)) \geq \sum_e \pi'(\ell(e)) \cdot \ell(e) \cdot \vartheta_1 = \sum_e \frac{\phi'(f(e))}{c^{-1}(e)} \cdot \frac{f(e)}{c(e)} \cdot \vartheta_1 = \Phi'(\mathbf{f}) \cdot \mathbf{f} \cdot \vartheta_1$$

Using, this, we obtain the following bound on the slackness (gap from optimum) $\aleph \equiv \dfrac{\Phi - \Phi^*}{\Phi}$

$$\aleph = \frac{\Phi - \Phi^*}{\Phi} < \frac{\Gamma - \Lambda}{\Phi} < \frac{\Gamma}{\Phi} \cdot (\xi + \tilde{\alpha}) < \frac{\Phi'(\mathbf{f}) \cdot \mathbf{f}}{\Phi(\mathbf{f})} \cdot (\xi + \tilde{\alpha}) < \frac{\xi + \tilde{\alpha}}{\vartheta_1} = \frac{\aleph_\dagger}{4} + \frac{\aleph_\dagger}{4} = \frac{\aleph_\dagger}{2} \tag{24}$$

Thus, $\xi \leq \vartheta_1 \cdot \aleph_\dagger / 4$ implies $\aleph < \aleph_\dagger / 2$, as required. (It is easy to see that multiplicative approximation for exponential potential implies additive approximation for maximum load; details in the sequel.) □

It remains to prove that either are in aggregate equilibrium, in which case we have accomplished a good approximation per Eqn. 20, or else potential will come down quickly enough within time interval $\mathcal{T}$. To prove the latter claim we will be strongly leveraging Eqn. 21.Analysis of Potential drop is order of magnitude harder than the equilibrium proof above; we handle this from Section B and on.

B Dynamics of potential drop

B.1 Mileage Definitions

Mileage. We are going to define *online mileage* $\nabla_i^\gamma(e,t)$, *offline mileage* $\nabla_i^\lambda(e,t)$, *derivative mileage* $\nabla^{\phi'}(e,t)$, as the absolute value of the change in the online potential, offline potential, and derivative of the objective function, respectively.

$$\nabla_i^\gamma(e,t) = |\gamma_i(e,t) - \gamma_i(e,t-1)| \tag{25}$$

$$\nabla_i^\lambda(e,t) = |\lambda_i(e,t) - \lambda_i(e,t-1)| \tag{26}$$

$$\nabla^{\phi'}(e,t) = |\phi'(e,t) - \phi'(e,t-1)| \tag{27}$$

We also define total mileage by summing over all commodities.

$$\nabla^{\gamma}(e,t) = \sum_i \nabla_i^{\gamma}(e,t) \tag{28}$$

$$\nabla^{\lambda}(e,t) = \sum_i \nabla_i^{\lambda}(e,t) \tag{29}$$

$$\nabla^{\Gamma}(t) = \sum_i \nabla_i^{\gamma}(t) = \sum_e \nabla_i^{\gamma}(e,t) = \sum_e \sum_i \nabla_i^{\gamma}(e,t) \tag{30}$$

$$\nabla^{\Lambda}(t) = \sum_i \nabla_i^{\Lambda}(t) = \sum_e \nabla_i^{\lambda}(e,t) = \sum_e \sum_i \nabla_i^{\lambda}(e,t) \tag{31}$$

For a path, $\mathbf{A} \in \mathcal{P}_i$, we define

$$\nabla_i^{\Gamma}(\mathbf{A},t) = \sum_{e \in \mathbf{A}} \nabla_i^{\gamma}(e,t) \tag{32}$$

$$\nabla_i^{\Lambda}(\mathbf{A},t) = \sum_{e \in \mathbf{A}} \nabla_i^{\lambda}(e,t) \tag{33}$$

$$\nabla^{\phi'}(\mathbf{A},t) = \sum_{e \in \mathbf{A}} \nabla^{\phi'}(e,t) \tag{34}$$

B.2 Mileage-based Commodity partitioning

Consider a time interval $\mathcal{T}$ with initial time t_0. We define online and offline mileage ratios for commodity i as

$$\rho_i^{\gamma}(\mathcal{T}) = \frac{\nabla_i^{\Gamma}(\mathcal{T})}{\Gamma_i(t_0)} \qquad \rho_i^{\lambda}(\mathcal{T}) = \frac{\nabla_i^{\Lambda}(\mathcal{T})}{\Gamma_i(t_0)} \tag{35}$$

We can decompose non-equilibrium commodities $\mathcal{O}$ into the following categories, $\mathcal{M}^{\gamma}, \mathcal{M}^{\lambda}, \mathcal{I}$, called migrant, mobile, or consistently non-equilibrium:

$$\mathcal{O} = \mathcal{I} \bigcup \mathcal{M}^{\gamma} \bigcup \mathcal{M}^{\lambda} \tag{36}$$

$$\mathcal{M}^{\gamma} = \{i \mid \rho_i^{\gamma}(\mathcal{T}) > \alpha\} \tag{37}$$

$$\mathcal{M}^{\lambda} = \{i \mid \rho_i^{\lambda}(\mathcal{T}) > \frac{\alpha^2}{2}\} \tag{38}$$

Let $\xi^{\gamma}, \xi^{\lambda}, \xi^{\mathcal{I}}$ be fractions of Γ for migrant, mobile & consistently non-equilibrium commodities:

$$\xi^{\gamma} = \frac{1}{\Gamma} \cdot \sum_{i \in \mathcal{M}^{\gamma}} \Gamma_i \;\; ; \;\; \xi^{\lambda} = \frac{1}{\Gamma} \cdot \sum_{i \in \mathcal{M}^{\lambda}} \Gamma_i \;\; ; \;\; \xi^{\mathcal{I}} = \frac{1}{\Gamma} \cdot \sum_{i \in \mathcal{I}} \Gamma_i \quad (\text{ clearly } \; \xi = \xi^{\gamma} + \xi^{\lambda} + \xi^{\mathcal{I}} \;\;) \tag{39}$$

B.3 Basic properties

Lemma B.1 *The costs increase/decrease within a round at most by factor* $1 + \alpha/4$.

Proof. By construction, multiplicative flow increase/decrease causes potential to increase at most by $1 + \alpha/8$. The additive increase can contribute $1 + \alpha/8$ fraction. □

Comment: Lemma B.1 shows that profit margin of α is not been "degraded" too much as a result of concurrency. Specifically, multiplicative rule leads to a change of $\alpha/8$ on both sides, i.e., $\alpha/4$ in total, and another $\alpha/8$ fraction is lost as a result of additive rule. All together, only 3/8 fraction of profit margin is "degraded". Thus, concurrent re-routing is almost as efficient as sequential re-routing.

Definition B.1 Let the *serialization* of the execution be sequential ordering of events in each round, taken in arbitrary order.

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

Proof. Whenever a flow quantum is re-routed from a source path to a destination path, the net reduction in potential is approximately difference between cost of source minus cost of destination, times the amount of flow being sent. In each round, flow is sent from high-cost paths to low-cost paths, with the costs calculated at the *beginning* of the round. At the beginning of the round, the profit cost margin is at least α fraction of the total volume of the change. The cost metric is the derivative of the potential functions, $\phi'(\mathbf{A})$, for each path $\mathbf{A}$. During the round, only $\alpha/4$ relative changes in cost is possible, in either direction. Thus, the original profit margin of α measured at the beginning of the round can be degraded by at most $\alpha/2$ during the round, leaving us with a lower bound of $\alpha/2$ on the profit margin. □

Corollary B.3 While μ is fixed, potential Φ_μ only goes down.

Proof. Directly follows from Lemma B.2. □

B.4 Proving that migrant commodities burn potential

In this section, we prove Eqn. 84 of Theorem C.1.

Proof. Fix a commodity i, and a server $\mathbf{A}$, and a server $\mathbf{B}$. Whenever flow δf flow is re-routed from $\mathbf{A}$ to $\mathbf{B}$, the commodity potential at $\mathbf{A}$ is decreased and potential at $\mathbf{B}$ is increased. The drop $\Delta\Phi$ in the potential is roughly δf times difference of derivatives.

$$\Phi(t-1) - \Phi(t) \approx \delta f(\phi'(\mathbf{A}) - \phi'(\mathbf{B})) > \delta f \cdot \alpha\phi'(\mathbf{A}) \tag{40}$$

By definition of profit margin, the total gain in potential is at least $\alpha\phi'(\mathbf{A})$:

$$\gamma'(\mathbf{A}) = \sum_{a\in\mathbf{A}} \gamma'(f(a)) = \sum_{a\in\mathbf{A}} \phi'(f(a)) + \phi''(f(a)) \cdot f(a) \tag{41}$$

$$= \sum_{a\in\mathbf{A}} \phi'(f(a))(1 + \frac{\pi''(\ell(a)) \cdot \ell(a)}{\pi'(\ell(a)}) \leq \sum_{a\in\mathbf{A}} \phi'(f(a))(1 + \frac{1}{\vartheta_2}) = \phi'(\mathbf{A}) \cdot (1 + \frac{1}{\vartheta_2}) \tag{42}$$

Thus, the online mileage can be upper-bounded as a fraction of the overall potential reduction:

$$\nabla^\Gamma(t) \leq \delta f(\gamma'(\mathbf{A}) + \gamma'(\mathbf{B})) \leq 2\delta f \frac{1}{\vartheta_2}(\phi'(\mathbf{A}) + \phi'(\mathbf{B})) \leq 4\delta f \frac{1}{\vartheta_2}\phi'(\mathbf{A}) < \frac{4}{\vartheta_2 \cdot \alpha}\Delta\Phi(t) \tag{43}$$

In other words, we have just proved

$$\Delta\Phi(t) > \frac{\vartheta_2 \cdot \alpha}{4}\nabla^\gamma(t) \tag{44}$$

Aggregating Migrant commodities. For each one of the migrant commodities, each unit of mileage translates into reduction in potential as in Eqn 44. The mobile commodity mileage is at least $\rho^{\lambda} \cdot \Gamma_i$, and summation of Γ_i over such commodities yields factor of $\xi^{\gamma} \cdot \Gamma$, thus resulting in total online mileage of

$$\rho^{\gamma} \cdot \xi^{\gamma} \cdot \Gamma = \frac{\Gamma}{\Phi} \cdot \Phi \cdot \rho^{\gamma} \cdot \xi^{\gamma} > \aleph \cdot \Phi \cdot \rho^{\gamma} \cdot \xi^{\gamma} \tag{45}$$

Translating online mileage into potential yields bound in Eqn. 84 of Theorem C.1.

$$\begin{aligned} \Delta\Phi &> \frac{\vartheta_2}{4}\aleph \cdot \Phi \cdot \rho^{\gamma} \cdot \xi^{\gamma} = \frac{\vartheta_2 \cdot \alpha}{4} \cdot \aleph \cdot \Phi \cdot \rho^{\gamma} \cdot \xi^{\gamma} = (\xi^{\gamma}\aleph \cdot \Phi) \cdot (\vartheta_1 \cdot \frac{\alpha}{4} \cdot \alpha) && (46) \\ &= (\xi^{\gamma}\aleph \cdot \Phi) \cdot (\alpha^2 \cdot \vartheta_1) = (\xi^{\gamma}\aleph \cdot \Phi) \cdot \chi^{\gamma} && (47) \end{aligned}$$

□

B.5 Proving that mobile commodities burn potential

We prove here Eqn. 85 of Theorem C.1.

Proof. Regarding the offline mileage, we have

$$\begin{aligned} \lambda'(\mathbf{A}) &= \sum_{e \in \mathbf{A}} \lambda'(f(e)) = \sum_{e \in \mathbf{A}} \phi''(f(a)) \cdot g(e) = \sum_{e \in \mathbf{A}} \phi'(f(e)) \cdot \frac{\phi''(f(e)) \cdot g(e)}{\phi'(f(e))} && (48) \\ &= \sum_{e \in \mathbf{A}} \phi'(f(e)) \cdot \frac{\pi''(\ell(e)) \cdot \ell^*(e)}{\pi'(\ell(e))} < \sum_{e \in \mathbf{A}} \phi'(f(e)) \cdot \frac{\pi''(\ell(e)) \cdot |\mathbf{f}|}{\pi'(\ell(e))} < \sum_{e \in \mathbf{A}} \phi'(f(e)) \cdot \frac{1}{\vartheta_3} && (49) \end{aligned}$$

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

$$\begin{aligned} \nabla^{\lambda}(t) &\leq \delta f(t) \cdot (\lambda'(\mathbf{A}) + \lambda'(\mathbf{B})) < \frac{1}{\vartheta_3} \cdot \delta f(t) \cdot (\phi'(\mathbf{A}) + \phi'(\mathbf{B})) && (50) \\ &= \frac{1}{\vartheta_3} \cdot \frac{2}{\alpha} \cdot \Delta\Phi(t) = \frac{2}{\vartheta_3 \cdot \alpha} \cdot \Delta\Phi(t) && (51) \end{aligned}$$

Or in other words, we have just proved

$$\Delta\Phi(t) \geq \Phi\nabla^{\lambda}(t) \cdot \frac{\vartheta_3 \cdot \alpha}{2} \tag{52}$$

Aggregating Mobile commodities. For each one of the mobile commodities, each unit of mileage translates into reduction in potential as in Eqn 52.

The mobile commodity mileage is at least $\rho^{\lambda} \cdot \Gamma_i$, and summation of Γ_i over such commodities yields factor of $\xi^{\lambda} \cdot \Gamma$, thus resulting in total online mileage of

$$\rho^{\gamma} \cdot \xi^{\gamma} \cdot \Gamma = \frac{\Gamma}{\Phi} \cdot \Phi \cdot \rho^{\gamma} \cdot \xi^{\gamma} > \aleph \cdot \Phi \cdot \rho^{\gamma} \cdot \xi^{\gamma} \tag{53}$$

Translating total offline mileage into potential drop yields drop claimed in Eqn. 85 of Thm. C.1.

$$\begin{aligned} \Delta\Phi &> (\frac{\vartheta_3 \cdot \alpha}{2}) \cdot (\aleph \cdot \Phi \cdot \rho^{\lambda} \cdot \xi^{\lambda}) = (\aleph \cdot \Phi\xi^{\lambda}) \cdot \alpha\vartheta_3\rho^{\lambda} && (54) \\ &= (\aleph \cdot \Phi\xi^{\lambda}) \cdot \alpha^3\vartheta_3 = (\aleph \cdot \Phi\xi^{\lambda}) \cdot O({\vartheta_1}^3 \cdot \vartheta_3 = (\aleph \cdot \Phi\xi^{\lambda}) \cdot O({\vartheta_1}^3 \cdot \vartheta_3) = \aleph \cdot \Phi \cdot \xi^{\lambda} \cdot \chi^{\lambda} && (55) \end{aligned}$$

□

B.6 Consistently non-equilibrium commodities burn potential

In this section, we prove Eqn. 86 of Theorem C.1.

B.6.1 Existence of an anchor

Lemma B.4 Consider a sample space Ω and a probability function $\pi : \Omega \to R^+$. Furthermore, consider two functions $\chi : \Omega \to R^+$ and $\psi : \Omega \to R^+$. Denote by $\bar{\chi}$ and $\bar{\psi}$ the expectations of χ and ψ under the probability distribution π, namely

$$\bar{\chi} = \sum_{\omega \in \Omega} \chi(\omega) \cdot \pi(\omega) \text{ and } \bar{\psi} = \sum_{\omega \in \Omega} \psi(\omega) \cdot \pi(\omega) \tag{56}$$

Then, for every $\alpha < 1$ there exists $\omega^* \in \Omega$ such that

$$\chi(\omega^*) \leq (1+\alpha) \cdot \bar{\chi} \tag{57}$$

$$\psi(\omega^*) \leq \frac{2}{\alpha} \cdot \bar{\psi} \tag{58}$$

Proof. Let us make the following definitions and observations:

$$\mathcal{L} = \{\omega \in \Omega \mid \chi(\omega)) < (1+\alpha)\bar{\chi}\} \tag{59}$$

$$\pi(\mathcal{L}) = \sum_{\omega \in \mathcal{L}} \pi(\omega) \geq \frac{\alpha}{2} \tag{60}$$

$$\chi_{max}(\mathcal{L}) \leftarrow \max_{\omega \in \mathcal{L}} \chi(\omega) \leq (1+\alpha)\bar{\chi} \tag{61}$$

$$\psi_{min}(\mathcal{L}) \leftarrow \min_{\omega \in \mathcal{L}} \psi(\omega) \leq \frac{2}{\alpha}\bar{\psi} \tag{62}$$

Note that the last inequality in Eqn. 60 holds because the cumulative probability of such set is at least $\alpha/2$. If this would not be the case, then with more than $1-\alpha/2$ probability, χ would have value which is α above the average, namely $\bar{\chi}$ thus contradicting the definition of the average, since total contribution to average over same space must exceed $\bar{\chi}$:

$$\bar{\chi} \cdot (1+\alpha) \cdot (1-\alpha/2) = \bar{\chi}(1+\alpha/2-\alpha^2/2) > \bar{\chi} \tag{63}$$

The inequalities in Eqn. 62 follow from the the following argument. Since fractional size of $\mathcal{L}$ is at least $\alpha/2$, then the minimum value of ψ in this set cannot exceed $2/\alpha$ times the average $\bar{\psi}$. □

Theorem B.5 (Anchor theorem) For each commodity i which is less than $\rho_i^\lambda = \alpha^2/2$-mobile in an interval $\mathcal{T}$ with a start point t_0 there exists an anchor path $\mathbf{B}_i$, for whom:

$$\nabla^{\phi'}(\mathbf{B}_i, \mathcal{T}) \leq \frac{2 \cdot \rho_i^\lambda}{\alpha} \cdot \bar{\Gamma}_i(t_0) < \alpha \cdot \bar{\Gamma}_i(t_0) = \alpha \cdot \frac{\Gamma_i(t_0)}{d_i} \tag{64}$$

$$\phi'(\mathbf{B}_i, t_0) < (1+\alpha) \cdot \bar{\Lambda}_i(t_0) = (1+\alpha) \cdot \frac{\Lambda_i(t_0)}{d_i} \tag{65}$$

Proof. Let consider the *average* offline potential, namely ratio of the total offline potential to total offline flow for a specific commodity.

By definition of mobility index ρ_i^λ, we conclude that the average derivative mileage, namely average of $\nabla^{\phi'}(\mathbf{B}, \mathcal{T})$, taken over all $\mathbf{B}$ w.r.t. volume measure of $g_i(\mathbf{B})$, is $\bar{\Gamma}_i \cdot \rho_i^\lambda$:

$$\begin{aligned}
\bar{\Gamma}_i \cdot \rho_i^\lambda &= \frac{1}{d_i}(\Gamma_i \cdot \rho_i^\lambda) = \frac{1}{d_i} \cdot \nabla_i^\lambda(\mathcal{T}) && (66)\\
&= \frac{1}{d_i} \cdot \sum_e \nabla^{\phi'}(e) \cdot g_i(e) = \frac{1}{d_i} \sum_e \nabla^{\phi'}(e) \cdot (\sum_{\mathbf{B}} g_i(\mathbf{B}, e)) && (67)\\
&= \frac{1}{d_i} \sum_{\mathbf{B}} \sum_e \nabla^{\phi'}(e) \cdot g_i(\mathbf{B}, e) = \frac{1}{d_i} \sum_{\mathbf{B}} g_i(\mathbf{B}) \cdot \sum_{e \in \mathbf{B}} \nabla^{\phi'}(e) && (68)\\
&= \frac{1}{d_i} \sum_{\mathbf{B}} g_i(\mathbf{B}) \cdot \nabla^{\phi'}(\mathbf{B}) && (69)
\end{aligned}$$

Similarly, note that $\bar{\Lambda}_i$ is the average value of $\phi'(\mathbf{B})$ taken over all $\mathbf{B}$ w.r.t. weighting by $g_i(\mathbf{B})$:

$$\bar{\Lambda}_i = \frac{\Lambda_i}{d_i} = \frac{1}{d_i} \sum_{\mathbf{B}} \lambda_i(\mathbf{B}) = \frac{1}{d_i} \sum_{\mathbf{B}} g_i(\mathbf{B}) \cdot \phi'(\mathbf{B}) \tag{70}$$

We thus can apply Lemma B.4, and choose anchor as $\mathbf{B}_i = \omega^*$

$$\mathbf{B}_i = \text{ argmin } \nabla^{\phi'}(\mathbf{A}, \mathcal{T}) \text{ over } \quad \mathbf{A} \in \mathcal{L}_i \tag{71}$$

which satisfies both of the bounds in Eqn. 62 and 61. Thm. B.5 follows. □

Corollary B.6 During the interval $\phi'(\mathbf{B}, t)$ can never exceed the following upper bound:

$$\phi'(\mathbf{B}_i, t) < \nabla^{\phi'}(\mathbf{B}_i, \mathcal{T}) + \phi'(\mathbf{B}_i, t_0) < (1 + \frac{\alpha}{2}) \cdot \bar{\Lambda}_i(t_0) + \frac{\alpha}{2}\bar{\Gamma}_i(t_0) \leq \bar{\Lambda}_i(t_0) + \alpha \cdot \Gamma_i(t_0) \tag{72}$$

Proof. Since anchor path $\mathbf{B}_i$ as stated in Thm. B.5 exists, note that for $\mathbf{B}_i$, Eqn. 64 and 65 hold, namely $\phi'(\mathbf{B}_i) < (1 + \frac{\alpha}{2}) \cdot \bar{\Lambda}_i$ and $\nabla^{\phi'}(\mathbf{B}_i, \mathcal{T}) < \frac{\alpha}{2}\bar{\Gamma}_i$, which implies the desired claim. □

B.6.2 Anchor effect

Notation B.1 We define parameters $\tau_0, \tau_\dagger, \tau_\S$ as follows:

$$\begin{aligned}
\tau_0 &= \max_{e \in E} \log_{1+\beta} \frac{c(e)}{\ddot{f}(e)} \approx \frac{1}{\beta} \log \frac{c(e)}{\frac{\mu}{\log_2 m} \cdot \frac{c(e)}{k} \log_2(1+\alpha)} \approx \frac{\log k}{\beta} && (73)\\
\tau_\dagger &= \tau_0 \cdot 8 \cdot H = \frac{8 \cdot H \cdot \log k}{\beta} && (74)\\
\tau_\S &= \frac{\tau_\dagger}{\aleph_\dagger{}^2 \cdot \vartheta_1{}^4 \cdot \vartheta_3} \approx \frac{8 \cdot H \cdot \log k}{\beta \cdot \aleph_\dagger{}^2 \cdot \vartheta_1{}^4 \cdot \vartheta_3} = \frac{8 \cdot H \cdot \log k}{\frac{\aleph_\dagger \cdot \vartheta_1}{4} \cdot \vartheta_2 \cdot \aleph_\dagger{}^2 \cdot \vartheta_1{}^4 \cdot \vartheta_3} = \frac{H}{\aleph_\dagger{}^3} \cdot \frac{\log k \cdot \log^7 m}{\epsilon^7} && (75)
\end{aligned}$$

Note that in definition of τ_0 Eqn. 73, the value maximized over all edges e is independent of e.

Note that in order to increase the value of the flow from $\ddot{f}(e)$ to $c(e)$, thus saturating edge e, it takes at most τ_0 time. The other parameter $\tau_\S$ is an upper bound on the time to complete a phase, as will be explained in Theorem C.2.

Theorem B.7 In any interval of length $\tau_\dagger$ as in Eqn. 74 or more, each consistently non-equilibrium commodity i contributes potential reduction of at least

$$\Delta\Phi > \Gamma_i \cdot \alpha \tag{76}$$

Proof. The proof proceeds by considering the following cases.

Case 1: If at any time t we manage to push $\frac{\beta}{4\cdot H}$ fraction of the demand on paths that are $1-\alpha$ cheaper than average, subject to the speed limit, then obviously potential comes down by factor

$$\frac{\alpha\cdot\beta}{4\cdot H}\cdot\Gamma_i \tag{77}$$

If this happens at least 25% of the time during interval of length $\frac{4\cdot H}{\beta}$ then the total decrease is by factor $\alpha\cdot\Gamma_i$.

Case 2: If on other hand, this event happens less than least 25% of the time during interval of length $\tau_\dagger$, then we get a contradiction as follows; total amount of saturated times is at least $1/4\tau_\dagger$. It must be the case that at each time during this time interval, the anchor path $\mathbf{B}_i$ must be have at least one saturated edge. It follows that for at least $\frac{\tau_\dagger}{4\cdot H}$ rounds, some edge e^* has been upward saturated. This leads to total increase by a factor of

$$(1+\beta)^{(\frac{\tau_\dagger}{4\cdot H})} \approx e^{\frac{\tau_\dagger\cdot\beta}{4\cdot H}} \tag{78}$$

Consider now the flow $f_i(e)$ over edges on the anchor path $e\in\mathbf{B}_i$. Such edge can suffer a reduction in the flow by a factor of at most $1-\frac{\beta}{8\cdot H}$ at each round, or cumulative reduction by factor of

$$(1-\frac{\beta}{8\cdot H})^{\tau_\dagger} \approx e^{-\frac{\tau_\dagger\cdot\beta}{8\cdot H}} \tag{79}$$

The product of these two terms is lower-bounded by

$$e^{\frac{\tau_\dagger\cdot\beta}{16\cdot H}} \approx (1+\beta)^{\frac{\tau_\dagger}{8\cdot H}} \tag{80}$$

Note that the highest increase possible is from $\ddot{f}_i(e)$, which is the lowest value on an edge e, to highest possible value, namely $c(e)$. Thus, the above lower bound must be smaller than the maximum possible increase, namely

$$(1+\beta)^{\frac{\tau_\dagger}{8\cdot H}} < \frac{c(e)}{\ddot{f}_i(e)} \quad \text{or} \quad \frac{\tau_\dagger}{8\cdot H} < \log_{1+\beta}\frac{c(e)}{\ddot{f}_i(e)} = \tau_0 \approx \frac{\log k}{\beta} \tag{81}$$

Eqn. 81 contradicts our choice of $\tau_\dagger$ in Eqn. 73, thus ruling this case out. □

To aggregate the effects of all the commodities, we can use the proof in Sec. B.6.3.

B.6.3 Aggregating non-equilibrium commodities

Each one of the consistently non-equilibrium commodities with online potential Γ_i generates reduction in potential as in Eqn 77. The mobile commodity has potential Γ_i, and summation of Γ_i over such commodities yields factor of $\xi^I\cdot\Gamma$, thus resulting in total non-equilibrium potential of

$$\xi^{\mathcal{I}}\cdot\frac{\Gamma}{\Phi}\Phi > (\Phi\cdot\aleph\cdot\xi^{\mathcal{I}}) \tag{82}$$

Translating non-equilibrium potential into objective function reduction potential as in Eqn 76 yields reduction in potential as claimed in Eqn. 86:

$$\Delta\Phi \quad > \quad (\Phi \cdot \aleph \cdot \xi^{\mathcal{I}}) \cdot (\vartheta_3 \cdot {\vartheta_1}^2) = (\xi^{\mathcal{I}} \cdot \Phi \cdot \aleph) \cdot \chi^{\mathcal{I}} \tag{83}$$

C Final running times bounds

C.1 Running time bounds for a phase

Notation C.1 We will use the following notations:

- *Beginning* of a phase is the time when value μ has been set for the last time.
- $\mathbf{f_0}$: flow at the beginning of a phase; $|\mathbf{f_0}| = \max_e \frac{\mathbf{f_0}(e)}{c(e)}$ is the maximum load of this flow.
- $\mathbf{f}_\dagger$: flow upon end of last phase; $|\mathbf{f}_\dagger| = \max_e \frac{\mathbf{f}_\dagger(e)}{c(e)}$ is the maximum load of this flow.
- $\mathbf{g} =$ argmin $|\mathbf{g}|$ is the optimum flow minimizing load and $\Phi^* = \Phi(\mathbf{g})$ is its potential value.
- $\aleph_\dagger < 1$ is the desired approximation of the potential compared to potential Φ^* of optimum flow.

Remark 1: The last phase does not need to terminate.

Remark 2: By end of the phase load is halved, $|\mathbf{f}_\dagger| < |\mathbf{f}|/2$ thus causing a change in μ.

Theorem C.1 Within a time interval $\mathcal{T} = (t_0, t_3)$ that is entirely included in one phase, the following potential improvements take place. (For Eqn. 86 to hold, length of $\mathcal{T}$ has to be at least $|\mathcal{T}| = \Omega(\tau_\dagger)$).

$$\frac{\Delta\Phi(\mathcal{T})}{\Phi(t_0)} \quad > \quad \frac{\Gamma(t_0)}{\Phi(t_0)} \cdot \xi^\gamma(\mathcal{T}) \cdot \chi^\gamma > \aleph \cdot \xi^\gamma(\mathcal{T}) \cdot \chi^\gamma \quad \text{where} \quad \chi^\gamma = \Omega({\vartheta_2}^2 \cdot \vartheta_1) \tag{84}$$

$$\frac{\Delta\Phi(\mathcal{T})}{\Phi(t_0)} \quad > \quad \frac{\Gamma(t_0)}{\Phi(t_0)} \cdot \xi^\lambda(\mathcal{T}) \cdot \chi^\lambda > \aleph \cdot \xi^\lambda(\mathcal{T}) \cdot \chi^\lambda \quad \text{where} \quad \chi^\lambda = \Omega({\vartheta_1}^3 \cdot \vartheta_3) \tag{85}$$

$$\frac{\Delta\Phi(\mathcal{T})}{\Phi(t_0)} \quad > \quad \frac{\Gamma(t_0)}{\Phi(t_0)} \cdot \xi^{\mathcal{I}}(\mathcal{T}) \cdot \chi^{\mathcal{I}} > \aleph \cdot \xi^{\mathcal{I}}(\mathcal{T}) \cdot \chi^{\mathcal{I}} \quad \text{where} \quad \chi^{\mathcal{I}} = \Omega(\vartheta_3 \cdot {\vartheta_1}^2) \tag{86}$$

Proof. Eqn. 84 of Theorem C.1 follows from Section B.4. Eqn. 85 of Theorem C.1 follows from Section B.5. Eqn. 86 of Theorem C.1 follows from Section B.6. □

Theorem C.2 Assuming the phase has not terminated $\tau_\dagger$ time since its start, the following must hold

- the current flow $\mathbf{f}$ satisfies $\Phi^* > (1 - \aleph_\dagger) \cdot \Phi(\mathbf{f})$.
- $|\mathbf{f}| < |\mathbf{g}| \cdot (1 + 2 \cdot \epsilon)$, i.e., we have achieved a flow with a near-optimal load.

Proof. If after $\tau_\dagger$ time after beginning of a phase, new phase has not started yet, then at that time $2 \cdot |\mathbf{f}_\dagger| > |\mathbf{f_0}|$ holds. This interval is long enough to apply Thm. C.2, implying large potential drop.

Recall that $\aleph = \dfrac{\Phi - \Phi^*}{\Phi} < \dfrac{\Gamma - \Lambda}{\Phi} < \dfrac{\Gamma}{\Phi}$. Since once we reach $\aleph = \aleph_\dagger$ we are done, we may as well assume that $\aleph > \aleph_\dagger$, in which case we can use Eqn. 21. In this case, potential drop is at least

$$\begin{aligned}
\frac{\Delta\Phi}{\Phi} &> \frac{\Gamma}{\Phi}(\chi^\gamma + \chi^\lambda + \chi^{\mathcal{I}}) > \aleph \cdot (\chi^\gamma \cdot \xi^\gamma + \chi^\lambda \cdot \xi^\lambda + \chi^{\mathcal{I}} \cdot \xi^{\mathcal{I}}) && (87)\\
&> \aleph \cdot (\xi^\gamma + \xi^\lambda + \xi^{\mathcal{I}}) \min\{\chi^\gamma, \chi^\lambda, \chi^{\mathcal{I}}\} = \aleph \cdot \xi \cdot \chi^\lambda > \Omega(\aleph \cdot \aleph_\dagger \cdot \vartheta_1 \cdot {\vartheta_1}^3 \cdot \vartheta_3) && (88)\\
&= \Omega({\aleph_\dagger}^2 \cdot {\vartheta_1}^4 \cdot \vartheta_3) && (89)
\end{aligned}$$

Thus, one of the two events happens in time $\tau_\S$: either phase terminates, with $|\mathbf{f}|$ dropping by a factor of 2, or, $\aleph_\dagger$ approximation for Φ is achieved. In the latter case, consider optimal flow $\mathbf{g}$ minimizing $|\mathbf{g}|$, for which

$$\begin{aligned}
m \cdot m^{\frac{|\mathbf{g}|}{\mu}} &= m \cdot \pi(|\mathbf{g}|) \geq \Phi^* \geq (1 - \aleph_\dagger) \cdot \Phi \geq (1 - \aleph_\dagger) \cdot \pi(|\mathbf{f}_\dagger|) = (1 - \aleph_\dagger) \cdot m^{\frac{|\mathbf{f}_\dagger|}{\mu}} && (90)\\
|\mathbf{f}_\dagger| &\leq |\mathbf{g}| + \mu \cdot (1 - \log_m(1 - \aleph_\dagger)) \approx |\mathbf{g}| + \mu \leq |\mathbf{g}| + |\mathbf{f_0}| \cdot \epsilon \leq |\mathbf{g}| + 2 \cdot |\mathbf{f}_\dagger| \cdot \epsilon && (91)
\end{aligned}$$

Eqn. 91 above implies $|\mathbf{g}| \geq |\mathbf{f}_\dagger|(1 - \epsilon)$. This concludes the proof of Theorem C.2. □

C.2 Scaling and bounds on the number of phases

Corollary B.3 & Thm. C.2, imply a number of corollaries.

Lemma C.3 At all times, we have $|\mathbf{f}| \leq |\mathbf{f_0}| + \mu$.

Proof. Note that additive increase in maximum load by μ, namely $|\mathbf{f}| > |\mathbf{f_0}| + \mu$ implies multiplicative increase in potential by m, i.e., since the initial potential is at most

$$m \cdot \phi(|\mathbf{f_0}|) = m \cdot m^{\frac{|\mathbf{f_0}|}{\mu}} \tag{92}$$

and the final potential is strictly larger, as it is lower bounded by

$$\phi(|\mathbf{f}|) = m^{\frac{|\mathbf{f}|}{\mu}} > m^{\frac{|\mathbf{f_0}|+\mu}{\mu}} = m^{\frac{\mu}{\mu}} \cdot m^{\frac{|\mathbf{f_0}|}{\mu}} \tag{93}$$

This is contradicting the fact that potential only coming down as claimed in Lemma B.3. □

Corollary C.4 At all times, we have

$$|\mathbf{f}| < |\mathbf{f_0}| + \mu < |\mathbf{f_0}|(1 + \epsilon) \tag{94}$$

Corollary C.5 At all times and for all edges $e \in E$, we have

$$\begin{aligned}
\phi(e) &< m^{1+\frac{1}{\epsilon}} \text{ and} && (95)\\
\phi'(e) &< \frac{\log m}{\mu} \cdot m^{1+\frac{1}{\epsilon}} && (96)
\end{aligned}$$

Lemma C.6 The total number of phases is at most $O(\log m)$.

Proof. Obviously, choosing the initial assignment as max-flow assignment for each commodity achieves maximum load $|\mathbf{f}| \leq |\mathbf{g}| \cdot O(k) = O(|\mathbf{g}| \cdot n^2)$. Note that the outer loop reduces the overload $|\mathbf{f}|$ by factor of 2 at least. □

Corollary C.7 $1 + \epsilon$ approximation is achieved by in total time $\tau_{\S} = O(\tau_{\dagger} \cdot \log n)$.