

Drift-Plus-Penalty Algorithm for Stochastic Network Optimization

Reference:

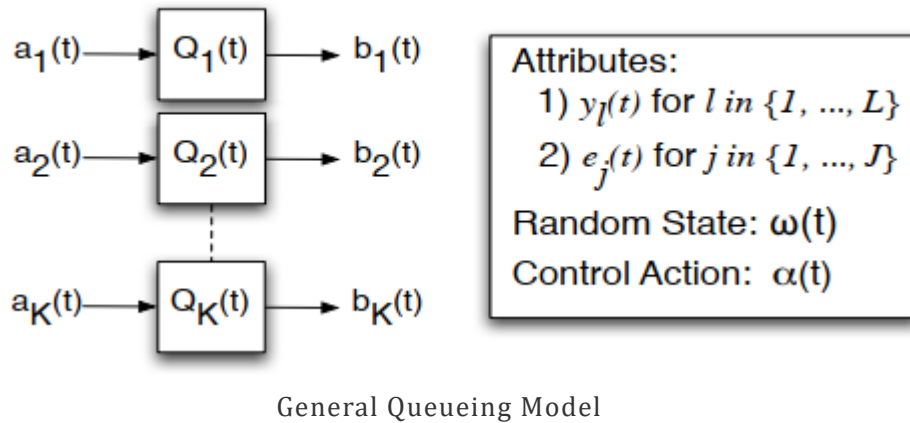
Neely, Michael J. "Stochastic network optimization with application to communication and queueing systems." Synthesis Lectures on Communication Networks 3.1 (2010): 1-211.

今天简单介绍一个最初在communication network发展出来的routing和resource allocation的算法，叫back pressure，是一个可以通过localized control来实现global optimality的一个在flow以及queueing network中应用非常广泛的算法和理论。参考文献对这个方法有非常详细的介绍，作者Michael J. Neely (from USC)很多的相关工作都是围绕这个算法以及相关的理论展开。这里是这位作者的[主页](#)。他的个人主页里面有很多相关的资料。这也是我一直以来认为美国学术圈很有意思我也比较喜欢的一个特点，很多老师做的研究都非常具有个人特色，每个人都有相对独立且研究比较深入的方向。

这个算法用一句话概括就是, We can use backpressure to achieve the **global queueing stability** using a **localized** routing and resource allocation policy in a **stochastic queueing network**. 后面我们就会具体展开介绍，这里怎么定义的**stability**，以及怎么用**一个localized policy**来实现全局网络的稳定。

Queue Stability and General Formulation

通用的模型大概可以用下图概括：



Queue length 每个slot的更新公式是：

$$Q_k(t+1) = \max[Q_k(t) - b_k(\alpha(t), \omega(t)), 0] + a_k(\alpha(t), \omega(t))$$

从这公式可以看出来，整个模型是建立在一个stationary random event $\omega(t) \sim \pi$ 的假设下，比如会假设arrival是i.i.d.的，这里 $\alpha(t)$ 是控制策略，要服从一系列的限制条件，arrival $a(t)$ 和departure $b(t)$ 都受到random event和控制策略的影响。

我们一般会假设arrival是每个时刻i.i.d.的，更具体的会假设是一个泊松过程，这里把 $a(t)$ 也写成 $\alpha(t)$ 的 function，主要不是为了表达达到的过程是受到控制策略影响，而是对于general network而言，下游的arrival就是上游的departure，这样写便于后面推广到general network (multi-hop in communication network and general signalized traffic network)，事实上本文不会介绍怎么推广到general network，这一步比较straight-forward。

Queue stability是为了描述整个network中排队稳定性的判据，最常用的判据也是最强的判据，叫**strongly stable**:

$$\limsup_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}|Q(\tau)| < \infty$$

简言之，就是系统内排队总长度，随着时间变化过程中，在infinite horizon的情况下是有界的。还有一个更松一点的稳定性，叫**mean rate stable**,这里也简述一下，相对而言理解第一个就可以了：

$$\lim_{t \rightarrow \infty} \frac{\mathbb{E}|Q(t)|}{t} = 0$$

在最开始的参考文献中，作者propose了下述的通用的optimization formulation，本算法基本可以认为是建立在解决这样一个问题的框架下：

$$\min \bar{y}_0$$

subject to:

1. $\bar{y}_l \leq 0 \quad \forall l$;
2. Queues $Q_k(t)$ are mean rate stable (strongly stable) $\forall k \in \{1, \dots, K\}$;
3. $\alpha(t) \in \mathcal{A}_{\omega(t)}$

目标函数是一个general的time average value，这个可以是我们的network中某个状态，也可以是某个状态的函数，限制条件是 1) some inequality constraints for some time averages; 2) Guarantee the queue stable; 3) action belongs to the allowed action space。还是非常好理解的，举一个简单的例子：在保证network中排队稳定的情况下最小化传输数据的功耗。

- REMARK 1 FOR INEQUALITY CONSTRAINTS: 后面将不会具体展开关于inequality constraints的介绍，we can put the inequality constraints to the *queue stability constraints* by converting the inequality constraints to a so-called **virtual queue**.
- REMARK 2: 这里的处理还是有一点反常规的，对于通常理解的，我们目标就是最小化延误，或者maximize throughput，我们把这里叫做效率上的保证，这里优化问题的formulation严格意义上讲有两个objectives，一个是保证效率，另一个是在保证效率的前提下优化其他（默认除效率之外的），后面的DPP也是一个trade-off。如果我们只看efficiency，那么这个优化问题就变成一个feasibility problem，我们把目标函数定成constant 0就行了，而backpressure policy就相当于这个优化问题feasibility problem的解。
- REMARK 3: 我们可以看到这里所有的对于效率的要求弱化成了一个稳定性要求，stability并不能完全体现效率，也就是说即便稳定了，但是并不是最好的，我们这里的处理设定非常类似于**optimal control**，但是并没有去做optimal control，所以其实也许可以用**reinforcement learning/optimal control**之类的方法论从另一个方面研究这个问题，但是从这些角度去研究的话，理论上很难去得到一些比较elegant的结论。

ω -only Polity and Admissible Demand Region

这里我们通过介绍 ω -only policy 来引出两个概念：**admissible demand region** 以及 **throughput-optimal**.

假定我们每一时刻可以observe random event 并且根据random event制定相应的策略，也就是说action是random event的函数： $\alpha(\omega(t))$ ，我们把这种policy叫做 ω -only policy。

Theorem 1: 如果原优化问题feasible (there exists a policy that can stabilize the system queues)，那么一定有一个 ω -only policy也可以是feasible solution。

这个证明有些繁琐，就不展开了，详见文初参考文献（实际上这大概是我唯一还没完全看明白的一个证明qaq）。这个theorem的好处是我们可以反过来用 ω -only policy去找 admissible demand region, 也即什么时候network queue可以被stablized，一来如果 ω -only policy 如果stablize不了的Queue，其他的算法也做不到(Theorem 1)，二来是 ω -only policy 形式很简单，因为我们直接可以通过取期望把唯一的pdf积分积出来：

$$\mathbb{E}\{a(\alpha(\omega(t)), \omega(t))\} = \bar{a} \quad \mathbb{E}\{b(\alpha(\omega(t)), \omega(t))\} = \bar{b}$$

这里直列了两个参数，整个系统所有的time average都可以直接取期望得到一个均值。Theorem 1可以用数学形式表达为，如果原优化问题有解的话，那么 $\forall \delta > 0$ ，存在一个 ω -only policy，使得：

$$\bar{y}_0 \leq y_0^{opt} + \delta, \quad \bar{a} \leq \bar{b} + \delta$$

这个条件我们后面会用到，用于证明network mean rate stable，比这个条件稍微更严格一点点的，我们叫Slater's condition，

$$\bar{y}_0 = \Phi(\epsilon), \quad \bar{a} \leq \bar{b} - \epsilon, \epsilon > 0$$

这个名字和convex optimization中的Slater's condition是一个含义，是为了ensure原问题有strictly feasible的解，我们之前也提到过两种不同的稳定性，对应起来看如果原优化问题是strictly feasible的，那么这个network就存在一个 ω -only policy使得network是strongly stable的，如果只是feasible的话，就只能保证network是mean rate stable了。

回过来说**admissible demand region**和**throughput-optimal**这两个概念，定理1最大的意义在于可以用一个比较简单的 ω -only policy去定义network的承载能力**capacity**，可以理解为**admissible demand region**的大小，在这个region内（如果fix routing的情况下

这个region是一个path flow的convex hull)。 **throughput-optimal**，是对一个算法的评价，我们称一个算法是**throughput-optimal**的如果这个算法可以stabilize the queueing network as long as there exists an algorithm/policy that could。就像我们之前提到的，这个**optimal**并没有想象的那么optimal，他只是稳定性判据，熟悉控制理论的应该都知道稳定性是非常基础的判据了。

- **EXAMPLE:** 感觉这一段讲的过于抽象了，这里举个例子阐释一下什么是 ω -only policy以及 **admissible demand region**，在Varaiya的traffic signal control的文章中，他把 ω -only policy用**fixed-timing plan**来解读，我觉得这个是一个很贴切精准的概念的迁移，在信号灯控制中这个所谓的 ω -only policy中就是我们假如已知了所有的routing/demand/saturation flow等全部的信息之后的固定配时的控制。在这样的设定下，admissible demand region也会很好理解，比如一个路口，东西方向和南北方向不能同时放行，那么东西方向与南北方向的总车流就不能超过这个路口的通行能力，否则就无法satisfy这个demand了，而且明显fixed routing 之后admissible demand region定义下的path flow 是一个convex hull.
- **REMARK:** 我的理解是 ω -only policy是一个数学形式简单，物理含义清晰的一个control policy，但是他最大的问题是我们去期望的过程中意味着我们需要知道 $\omega(t) \sim \pi$ 的具体分布形式，这个并不是特别容易获得，尤其是对于一个general network而言，这个要求的是全局的stochastic distribution，是个很强的假设，比如全局的routing和demand的信息之类的。

Lyapunov Stability and Optimization Theorem

对于一个queueing network，我们定义Lyapunov function 以及Lyapunov drift如下：

$$L(\mathbf{Q}(t)) = \frac{1}{2} \sum_i Q_i^2(t) \quad \Delta(\mathbf{Q}(t)) = \mathbb{E}\{L(\mathbf{Q}(t+1)) - L(\mathbf{Q}(t)) \mid \mathbf{Q}(t)\}$$

我尽量用粗体区分矢量与标量，或者如果没有下标的一般会指矢量。Lyapunov stability theorem主要是针对稳定性的（feasibility problem），Lyapunov Optimization Theorem则是更general的优化问题。

Theorem 2 (Lyapunov Stability): 对于一个queueing network，在之前的Lyapunov function以及drift的定义下，假定 $\mathbb{E}\{L(\mathbf{Q}(0))\} < \infty$ (boundness assumption，很显然一般都是成立的)。假定任意参数 $B > 0, \epsilon \geq 0$ ，下式成立：

$$\Delta(\mathbf{Q}(t)) \leq B - \epsilon \sum_i |Q_i(t)|, \quad \forall t$$

那么如果 $\epsilon = 0$, network就是mean rate stable的, 如果大于零, 则strongly stable。这个证明非常简单, 对于每个slot t 这个不等式都成立, 把他们累加起来就可以了。

Theorem 3 (Lyapunov Optimization): 对于一个queueing network, 在之前的Lyapunov function以及drift的定义下, 假定 $\mathbb{E}\{L(\mathbf{Q}(0))\} < \infty$ (boundness assumption, 很显然一般都是成立的)。假定任意参数 $B > 0, \epsilon \geq 0, V \geq 0$ and y^* , 下式成立:

$$\Delta(\mathbf{Q}(t)) \leq B + Vy^* - \epsilon \sum_i |Q_i(t)|, \quad \forall t$$

那么Queue就有mean rate stable, 进一步如果 $\epsilon > 0, V > 0$, 有:

$$\begin{aligned} \limsup_{t \rightarrow \infty} \frac{1}{t} \sum_t \mathbb{E}y(t) &\leq y^* + \frac{B}{V} \\ \limsup_{t \rightarrow \infty} \frac{1}{t} \sum_t \sum_i \mathbb{E}\{Q_i(t)\} &\leq \frac{B + V(y^* - y_{min})}{\epsilon} \end{aligned}$$

整体上这两个定理证明都是很简单的, 特别是对于strongly stable的证明, 基本就是不等式累加就可以得到。后面我们就会利用这个定理的性质, 去构造满足这个定理的条件来达到我们想要的目的。

Drift-Plus-Penalty Algorithm for Routing and Resource Allocation

这个章节我们带着之前花了很大力气准备的问题的formulation, ω -only policy以及在他定义下的admissible demand region以及Lyapunov stability theorem, 来看drift-plus-penalty是如何解决最开始我们propose的问题的。

Lyapunov Theorem 中的条件是给Lyapunov drift一个上界, 我们回到最开始的dynamic equation,

$$Q_k(t+1) = \max[Q_k(t) - b_k(\alpha(t), \omega(t)), 0] + a_k(\alpha(t), \omega(t))$$

我们把这一项两遍同时平方, 右边用 $(\max\{a - b, 0\})^2 \leq (a - b)^2$ 放缩一下, 然后通过简单的移项可以得到:

$$\frac{Q_k(t+1)^2 - Q_k(t)^2}{2} \leq \frac{a_k(t)^2 + b_k(t)^2}{2} - \tilde{b}_k(t)a_k(t) + Q_k(t)[a_k(t) - b_k(t)]$$

左边是Lyapunov drift, 右边我们回顾之前的Lyapunov theorem, 我们并不关心没有 $Q_k(t)$ 的项, 只要他们是有界的即可, 这样我们就有:

$$L(Q_k(t+1)) - L(Q_k(t)) \leq B + Q_k(t)[a_k(t) - b_k(t)]$$

我们两边同时加上 $Vy(t)$ 然后两边同时取 conditional expectation (given $\mathbf{Q}(t)$) 可以得到:

$$\Delta(\mathbf{Q}(t)) + V\mathbb{E}(y(t) \mid \mathbf{Q}(t)) \leq B + V\mathbb{E}(y(t)) + \sum_i Q_k \mathbb{E}[a_k(t) - b_k(t) \mid \mathbf{Q}(t)]$$

Minimize drift-plus-penalty algorithm: 就是 minimize 上面不等式 rhs 的算法, 严格意义上讲应该是 minimize drift-plus-penalty 的一个 upper bound, 完整写出来就是:

$$\min V y(\alpha(t), \omega(t)) + \sum_i Q_k \{a_k(\alpha(t), \omega(t)) - b_k(\alpha(t), \omega(t))\}$$

subject to $\alpha(t) \in \mathcal{A}\omega(t)$.

我们先看这个算法的性质, 随后证明这个算法为什么可以去解原先的优化问题:

- **REMARK 1:** 相对于 ω -only policy, 这里很重要的一个性质就是不需要知道 $\omega(t)$ 的具体分布, 只需要不停地观测当前状态 (以及 realized uncertainty) 然后制定策略, 比如不需要知道 demand 的具体信息;
- **REMARK 2:** 控制策略的目标函数分两项, 第一项是 penalty 项, 第二项是关于 drift 的一项, 所以叫 drift-plus-penalty, 后面会具体展开这个是个线性的 trade-off between the stability and minimizing the objective function in the original formulation。
- **REMARK 3:** backpressure algorithm. 这个推广到 multi-hop 可以拆解为 routing 和 resource allocation 两种结合的控制策略, 这里不具体展开。如果是 fixed routing 的话, 我们看第二项, 我们可以把 summation 换顺序把同一个 flow 合并同类项, 就得到了 max pressure 或者叫 back pressure 的 localized control policy.

针对这个算法的证明有了前面的铺垫是非常直接的, 这里我大致描述一下思路, 首先我们这里 minimize 了 drift-plus-penalty 的 upper bound, 也就是说我们得到了右侧 upper

bound的最低值，也就意味着我们前面铺垫的 ω -only policy能够achieve的upper bound是比这个大的，也就是说我们要把 ω -only policy拿过来放缩一步。我们把upper bound记为：

$$UB(\alpha(t), \mathbf{Q}(t)) = V\mathbb{E}(y(t) + \sum_i Q_k \mathbb{E}[a_k(t) - b_k(t) \mid \mathbf{Q}(t)])$$

我们后面缺省了upper bound的 $\mathbf{Q}(t)$ ，那么我们有：

$$\Delta(\mathbf{Q}(t)) + V\mathbb{E}(y(t) \mid \mathbf{Q}(t)) \leq UB(\alpha^{dpp}(t)) \leq UB(\alpha^*(t))$$

这里的 α^{dpp} 是dpp algorithm下的policy, $\omega^*(t)$ 是**optimal/feasible ω -only policy**，我们回到之前对 ω -only policy的定义以及他所定义的admissible demand region。如果当前的优化问题有解，我们就可以有：

$$\bar{y}_0 \leq y_0^{opt} + \delta, \quad \bar{a} \leq \bar{b} + \delta$$

或者更严格的：

$$\bar{y}_0 = \Phi(\epsilon), \quad \bar{a} \leq \bar{b} - \epsilon, \epsilon > 0$$

我们把这个bound加到 $UB(\alpha^*(t))$ 的最右边就会发现我们得到了Lyapunov Optimization Theorem 的条件一模一样的形式。

DPP for Traffic Signal Control

这个算法发展于communication network，但是并不局限于在通讯网络中，在2013年Pravin Varaiya院士把这个算法引入到了network traffic signal control中，随后就有一些列工作围绕这个工作进行展开，目前主要的问题是traffic network和communication network (data flow network) 模型上有一些本质性的差异，直接用原始的backpressure个人认为目前得到的都是一些intuition和basic ideas，更多的是理论上的一些价值离实际应用还是相去甚远。

- Comparision between Communication Network and Signalized Traffic Network:

Contents	Signalized Traffic Network	Communication Network
----------	----------------------------	-----------------------

Contents	Signalized Traffic Network	Communication Network
Control variables	Traffic signal/Traffic assignment	Resource allocation/routing
Queueing	Point-queue will undermine coordination	Point-queue
Switch loss	High	Ignored
Routing	Selfish routing (User equilibrium)	Controlled routing
Objective	Minimize total delay	Minimize power cost, queue stable

其中控制变量和目标函数两个部分是几乎相同的，模型整体上是相似的，都是一个 queueing 或者 flow network，这也是为什么通讯网络的理论发展被用到了信号灯控制里，但是有一些地方又是不同的，个人认为主要是数据网络相对而言，不需要考虑过多 link capacity 和 link travel time，但是在交通网络中非常关键，尤其是做协调控制，

有很多相关的参考文献，看明白两个我觉得就很够了，后面期待可以在这个 research path 上面对于这个算法真正能够落地可以做出一点努力：

- Varaiya, Pravin. "Max pressure control of a network of signalized intersections." Transportation Research Part C: Emerging Technologies 36 (2013): 177-195. [\[link\]](#)
- Li, Li, and Saif Eddin Jabari. "Position weighted backpressure intersection control for urban networks." Transportation Research Part B: Methodological 128 (2019): 435-461. [\[link\]](#)

DPP Algorithm for Convex Optimization

这个章节是一个相对独立与前面的章节，原书作者貌似不满足于这个算法应用于 communication network，不仅仅写了一本书针对于 general stochastic queueing network 吸引了我这种本来不做 communication network 的人了解了他的工作，甚至于还自己根据前面的算法设计了一个解 convex optimization 的问题，我更倾向于把他这个神奇的算法叫做一个 online 的解法。

写推送真的貌似有点累的样子，这个地方大家感兴趣的话参考：[Network Optimization: Notes and Exercies](#) 中第六部分。