Top-percentile traffic routing problem by dynamic programming

Grothey, A & Yang, X 2012, 'Top-percentile traffic routing problem by dynamic programming', Optimization and Engineering, vol. 12, no. 4, pp. 631-655

Top-percentile traffic routing problem by Dynamic Programming

Andreas Grothey, Xinan Yang
School of Mathematics
College of Science and Engineering
The University of Edinburgh

May 6, 2009

Abstract

Multi-homing is a technology used by Internet Service Provider (ISP) to connect to the Internet via different network providers. To make full use of the underlying networks with minimum cost, an optimal routing strategy is required by ISPs. This study investigates the optimal routing strategy in case where network providers charge ISPs according to top-percentile pricing. We call this problem the Top-percentile Traffic Routing Problem (TpTRP). The TpTRP is a multistage stochastic optimisation problem in which routing decision should be made before knowing the amount of traffic that is to be routed in the following time period. The stochastic nature of the problem forms the critical difficulty of this study.

In this paper several approaches are investigated in modelling and solving the problem. We begin by modelling the TpTRP as a multistage stochastic programming problem, which is hard to solve due to the integer variables introduced by top-percentile pricing. Several simplifications of the original TpTRP are then explored in the second part of this work. Some of these allow analytical solutions which lead to bounds on the achievable optimal solution. We also establish bounds by investigation several "naive" routing policies. In the end, we explore the solution of the TpTRP as a stochastic dynamic programming problem by a discretization of the state space. This allows us to solve medium size instances of TpTRP to optimality and to improve on any naive routing policy.

1 Introduction

Internet Service Providers (ISPs) do not generally have their own network infrastructure to route the incoming traffic of their customers, but instead use external network providers. Multi-homing is used by ISPs to connect to the Internet via more than one network provider (see Figure 1). This technique is becoming more and more popular in ISP organisations as it improves the reliability and quality of service of the ISP. When failure occurs in one of the networks or its quality degrades, the ISP can use an alternative network.

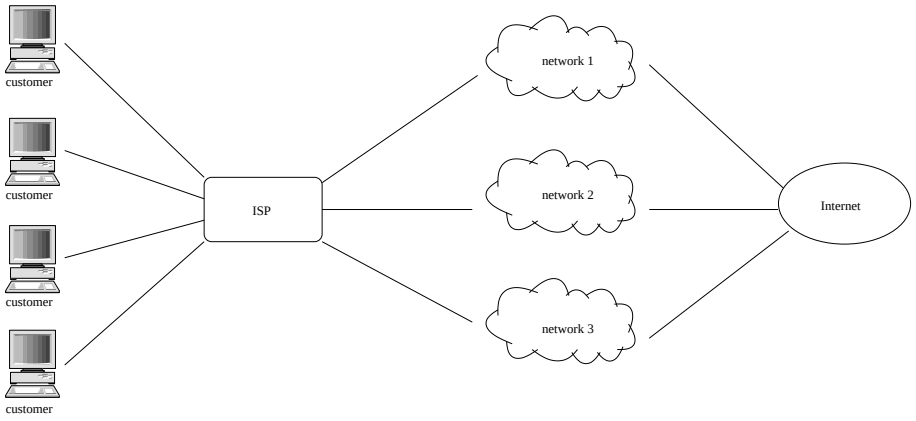

Figure 1: ISP connect customers to the Internet under multi-homing technique

However, an ISP that uses multi-homing is subject to extra charges due to the use of multiple networks. Important questions that are faced by such an ISP is how to assign traffic among network providers to minimize the inflicted cost of multi-homing. Of course, the answer to this question depends on the pricing policy used by network providers. In this work we consider top-percentile pricing.

Top-percentile pricing is a relatively new and increasingly popular pricing policy used by network providers to charge service providers, that is quickly becoming established in [7]. In this scheme, the network provider divides the charge period, say a month for example, into several time intervals with equal, fixed length. Then, it measures and evaluates the amount of data (traffic) sent in these time intervals. At the end of the charge period, the network provider select the traffic volume of the top q-percentile interval as the basis for computing the cost. For example, if the charge period (i.e. 30 days) is divided into 4320 time intervals with the length of 10 mins, and if top 5-percentile pricing is used, the cost computed by top-percentile pricing is based on the traffic volume of the top 216th interval.

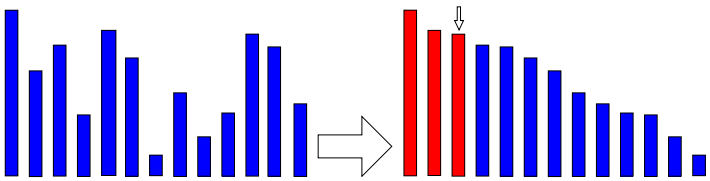

Figure 2: Random traffics, reordered traffics and the top-percentile on which charge are based

It is worthwhile to point out that, under top-percentile pricing, the ISP can ship several traffics via a network without being charged by its provider. For example, Figure 2 shows a simplified top-percentile pricing procedure, in which we have 10 time intervals in total and are charged according to the 3rd highest volume of shipped traffic. In this instance, every time interval gets a nonzero amount of traffic shipped. If the network provider charges c on per unit traffic shipped during the top-3rd time interval, then the cost amounts to cT^3. However, if the ISP send no more than 2 traffics to a network during the charge period, it will not be charged by this network provider as there is no traffic in the 3rd highest time interval. So that under pure top-percentile pricing policy, the ISP has a chance to not be charged by a network provider although it did send data with it.

Traditionally, network providers have used pricing policies such as fixed-cost pricing and per-usage pricing. Under per-usage pricing, the ISP pays for the actual amount of traffic shipped over the charge period, while fixed-cost pricing means the ISP pays a fixed price for the charge period regardless of the amount of traffic shipped. The optimal routing policies under these pricing mechanisms have been explored in the past literature. For example, Altman et al. [1] and Wang, Schulzrinne [11] investigate the competitive routing problem under per-usage pricing, while the fixed-cost price routing policy is investigated in [8], [5]. In contrast to the traditional pricing policies, very little work has been done on network design and operation under top-percentile pricing. The deterministic problem (in which we assume that we know all the volumes of traffic in advance) has been analysed in [3], where the authors build a mixed-integer linear programming model and develop an efficient B&B algorithm to solve it. For stochastic considerations, Levy et al. in [6] develops a probabilistic model and provides an analysis of the expected costs, thus demonstrate that multi-homing can be economical efficient. On the other hand, Goldenberg et al. [4] focus on the development of smart routing algorithms for optimising both cost and performance for multi-homing users. However, to the best of our knowledge there is no result dealing with the optimal multi-homing routing policy under top-percentile pricing.

The purpose of this study is to determine the optimal routing strategy under top-percentile pricing policy. Precisely, if all network providers charge

the ISP based on the volume of the top q-percentile time interval's traffic, how to allocate all time intervals' traffics among those networks to minimise the total cost charged on the ISP. In practise, network providers might combine top-percentile pricing with other pricing policies. In this work however, we investigate the optimal multi-homing routing policy in the context of pure top-percentile pricing, namely network providers charge the ISP according to and only according to the traffic volume on the top-percentile time interval. For the reason of clarity, in the following parts of this paper we call this problem, the Top-percentile Traffic Routing Problem (TpTRP). We assume that the ISP can not anticipate the volume of future traffics. Instead, what the ISP knows are the probabilistic distributions of every time intervals' traffic. Under this precondition, the TpTRP becomes a multi-stage stochastic problem in which decision should be made under significant uncertainty.

As a starting point, we model the TpTRP as a multi-stage stochastic programming problem. Unfortunately, the modelling of the top-percentile cost requires the introduction of integer variables within the last time stage. This makes the multi-stage stochastic programming model too difficult to be solved for any but the smallest instances. On the other hand, we suggest two simplifications of the model which are analytically solvable. The naive routing policies derived from them provide us with lower and upper bounds on the optimal solution of the TpTRP. Finally we suggest to solve the TpTRP as a stochastic dynamic programming problem by a discretization of the state space. This model is solvable for medium size instances, giving a routing policy which outperforms any naive routing policy. With some modifications, the decision rule concluded from this routing policy can be implemented on real-size instances as well and likewise outperforms any naive routing policy.

The remainder of this report is organised as follows. In Section 2 we introduce the multi-stage mixed-integer stochastic programming model arising from the TpTRP and demonstrate the difficulties in solving it. Section 3 is about the simplifications, where we derive several naive routing policies. We present the discretization step, build the stochastic dynamic programming model of TpTRP in Section 4. Then in Section 5, we numerically examine the quality and reliability of all the routing policies we produced by implementing them onto several instances. Finally, conclusions and future works are summarised in Section 6.

2 Stochastic Programming Model

2.1 Problem specification

- $I, |I| = n$: The set of network providers.
  There are n underlying network providers. We assume that all the network providers use top-percentile pricing with a same percentile parameter q. Network provider i charges c_i per unit of traffic transfered on the top-percentile interval. We assume further that there is no upper bound on the volume of traffic that can be transfered by each network provider, and no failure occurring in any network during the charge period.

- Γ : The set of time intervals.
  We assume that all network providers divide the charge period into the same $|\Gamma|$ time intervals of equal length.

- $\theta = \lfloor |\Gamma| * q \rfloor$: The index of the top-percentile time interval.
  Cost inflicted on the ISP charged by network provider i is a function of the traffic volume sent within the θ-th highest time interval by network provider i. Namely $Cost_i = c_i y_i$, if y_i is the θ-th highest volume of traffic shipped by network provider i.

- $T^\tau, \tau \in \Gamma$: The volume of traffic required to be sent in time interval τ.
  We assume that before the routing decision for period τ is made, $T^\tau(\omega^\tau)$ is a random variable depending on the random event ω^τ.

- $x^\tau = (x_1^\tau, x_2^\tau, ..., x_n^\tau)^T, \tau \in \Gamma$: The routing decision for time interval τ.
  At the beginning of every time interval, a routing decision x^τ should be made without knowing the amount of traffic T^τ. The question is, what are the possible choices of decision? For example, we can make decision on the fraction of traffic T^τ to be sent by every network provider. However, this is not the only possible choice.

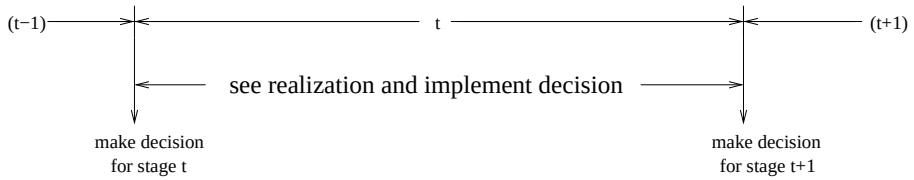

Figure 3: Process of data revelation and implementation of decisions

As shown in Figure 3, T^τ is the cumulative data sent during time interval τ, the complete amount of T^τ is not fully revealed until the end of the time interval. However, any traffic must be sent as soon as it is generated, according to the routing decision we have made. Therefore a necessary condition for a feasible routing decision is that it is implementable without knowing the volume of traffic. For example, we could route traffic up to a threshold amount with one network provider and split the remainder among the rest network providers according to some predetermined fractions.

2.2 Multi-stage mixed-integer stochastic programming model

Now we build the stochastic programming model for the TpTRP. We treat every time interval as a stage; during each of the $|\Gamma|$ stages, a routing decision has to be made on how to route that time interval's traffic with the available network providers. The decision x_i^τ is the fraction of traffic T^τ to be sent by network provider i: so that the volume of traffic for network i is $x_i^\tau T^\tau$.

Let $\omega_\tau = (\omega^1, \omega^2 ..., \omega^\tau)$ be the information available at time stage τ, i.e. the realisations of the traffic volumes $T^1, T^2, ..., T^\tau$. Further denote by $x_\tau = (x^1, x^2, ..., x^\tau)$ the history of routing decisions up to time stage τ. Let $Q^\tau(x_\tau, \omega_\tau)$ denote the value function at time stage τ, i.e. the expected cost of being at time stage τ with decision history x_τ and traffic realisations ω_τ. Then for stages $\tau = 0, ..., |\Gamma| - 1$, the TpTRP can be stated in recourse form as:

$$\begin{cases} Q^\tau(x_\tau, \omega_\tau) = \min\limits_{x^{\tau+1}} \quad \mathbb{E}_{\omega^{\tau+1}}(Q^{\tau+1}(x_{\tau+1}, \omega_{\tau+1})) \\ \qquad\qquad s.t.: \quad \sum\limits_{i \in I} x_i^{\tau+1} = 1, \\ \qquad\qquad\qquad\quad 0 \leq x_i^{\tau+1} \leq 1, \forall i \in I \end{cases}$$

For terminal conditions, given all the decisions up to time stage $|\Gamma|$, an additional stage $|\Gamma|$ is added to calculate the cost implied by the decision set $x_{|\Gamma|} = (x^1, x^2, ..., x^{|\Gamma|})$ under top-percentile pricing policy. We introduce binary variables $z_i^{\tau, \omega_{|\Gamma|}}$ to distinguish whether a particular time interval τ is in the top q-percentile time intervals shipped by network provider i, under the realisations $\omega_{|\Gamma|} = (\omega^1, \omega^2, ..., \omega^{|\Gamma|})$. With the help of these binary variables we can finally find the volume of the top-percentile traffic $y_i^{\omega_{|\Gamma|}}$ for network provider i, thus the cost charged on the ISP. For stage $\tau = |\Gamma|$,

$$\begin{cases} Q^{|\Gamma|}(x_{|\Gamma|}, \omega_{|\Gamma|}) = \min\limits_{z} \sum\limits_{i \in I} c_i y_i^{\omega_{|\Gamma|}} \\ \qquad\qquad s.t.: \quad \sum\limits_{\tau \in \Gamma} z_i^{\tau, \omega_{|\Gamma|}} \leq \theta, \forall i \in I \\ \qquad\qquad\qquad\quad x_i^{\tau} \omega^{\tau} - y_i^{\omega_{|\Gamma|}} \leq z_i^{\tau, \omega_{|\Gamma|}} \omega^{\tau}, \forall i \in I, \forall \tau \in \Gamma \\ \qquad\qquad\qquad\quad z_i^{\tau, \omega_{|\Gamma|}} \in \{0, 1\}, \forall i \in I, \forall \tau \in \Gamma \end{cases}$$

From the above discussion we can see that the last stage problem is just determining the volume of top q-percentile traffic for every network provider, i.e. the θth highest volume of $\{x_i^1 \omega^1, x_i^2 \omega^2, ..., x_i^{|\Gamma|} \omega^{|\Gamma|}\}$ for network provider i. Although this step seems easy conceptually, it has to be modelled using integer variables, because the $\theta-$th highest volume of $\{x_i^1 \omega^1, x_i^2 \omega^2, ..., x_i^{|\Gamma|} \omega^{|\Gamma|}\}$ is non-convex in its arguments.

In conclusion, we build a $(|\Gamma| + 1)$-stage mixed-integer stochastic programming model with binary requirement only on some of the last stage variables to model the TpTRP. Given a discrete set of random realisations $\Omega^{\tau}, \tau = 1, ..., |\Gamma|$, the deterministic equivalent of this multi-stage stochastic model is:

$$\begin{cases} obj = \min\limits_{x^1} \mathbb{E}_{\omega^1 \in \Omega^1} [\min\limits_{x^2} \mathbb{E}_{\omega^2 \in \Omega^2} [...[\min\limits_{x^{|\Gamma|}} \mathbb{E}_{\omega^{|\Gamma|} \in \Omega^{|\Gamma|}} (\min\limits_{z} \sum\limits_{i \in I} c_i y_i^{\omega_{|\Gamma|}})]...]] \\ s.t.: \quad \sum\limits_{i \in I} x_i^{\tau, \omega_{\tau}} = 1, \forall \tau \in \Gamma, \forall \omega_{\tau} \in \Omega^1 \times ... \times \Omega^{\tau} \\ \qquad\quad x_i^{\tau, \omega_{\tau}} \omega^{\tau} - y_i^{\omega_{|\Gamma|}} \leq z_i^{\tau, \omega_{|\Gamma|}} \omega^{\tau}, \forall i \in I, \forall \tau \in \Gamma, \forall \omega_{|\Gamma|} \in \Omega^1 \times ... \times \Omega^{|\Gamma|} \\ \qquad\quad \sum\limits_{\tau \in \Gamma} z_i^{\tau, \omega_{|\Gamma|}} \leq \theta, \forall i \in I, \forall \omega_{|\Gamma|} \in \Omega^1 \times ... \times \Omega^{|\Gamma|} \\ \qquad\quad 0 \leq x_i^{\tau, \omega_{\tau}} \leq 1, \forall i \in I, \forall \tau \in \Gamma, \forall \omega_{\tau} \in \Omega^1 \times ... \times \Omega^{\tau} \\ \qquad\quad z_i^{\tau, \omega_{|\Gamma|}} \in \{0, 1\}, \forall i \in I, \forall \tau \in \Gamma, \forall i \in I, \forall \tau \in \Gamma, \forall \omega_{|\Gamma|} \in \Omega^1 \times ... \times \Omega^{|\Gamma|} \end{cases} \tag{2.1}$$

Problem (2.1) is a standard mixed-integer stochastic programming problem. If we assume there are K discrete realizations $\omega^{\tau} \in \Omega^{\tau}$ per stage ($K = |\Omega^{\tau}|, \tau = 1, ..., |\Gamma|$), then the problem has $(\sum\limits_{\tau=1}^{|\Gamma|} K^{\tau - 1} + |I| \cdot K^{|\Gamma|} + |I| \cdot |\Gamma| \cdot K^{|\Gamma|})$ constraints and $|I| \cdot (\sum\limits_{\tau=1}^{|\Gamma|} K^{\tau-1} + K^{|\Gamma|} + |\Gamma| \cdot K^{|\Gamma|})$ variables of which $|I| \cdot |\Gamma| \cdot K^{|\Gamma|}$ are integer.

2.3 Numerical results and difficulties

The simplest approach to solve the problem is to consider its deterministic equivalent as a large scale monolithic mixed-integer linear program and to apply a commercial standard solver, e.g. CPLEX, on it. Unfortunately,

the size and the integer nature of the problem prevents the efficient use of standard commercial software, even on fairly small instances.

For example, in Instance 1 we assume that we have 2 potential network providers to choose from, both of them divide the charge period into 5 time intervals and charge based on the 2nd highest volume of traffic. Every time interval's traffic follow a same discrete distribution, with a mean value 8000. At every time stage τ, the potential realisation set $\Omega^\tau = \{6000, 8000, 10000\}$ and $pr(6000) = pr(8000) = pr(10000) = 0.33$. This leads to a stochastic tree with $3^5 = 243$ scenarios and a deterministic equivalent problem size of 3037 constraints and 3158 variables, in which 2430 are restricted to be binary.

Instead of trying to solve this problem we make a simplification on the decision structure. We combine the first $|\Gamma|$ stages together, making all decision indifferently within the first time stage, and then compute the cost in the final stage. This simplified two-stage model reduces the size of the deterministic equivalent problem, however, it still need to examine $3^5 = 243$ scenarios in total. In the deterministic equivalent, we get 10 first stage variables and 2916 second stage variables, in which 2430 are restricted to be binary. With 2921 constraints, this deterministic equivalent of the two-stage mixed-integer stochastic model could not be solve to optimality by CPLEX. (Elapsed time = 344722.09 sec. (tree size = 773.62 MB). Gap = 37.88%)

Although Instance 1 is a fairly small instance of the TpTRP, its deterministic equivalent is too large to be solved by commercial solver such as CPLEX. In fact, multi-stage stochastic programming problems are well known for being challenging both from theoretical and computational points of view, because they can lead to very large scale problems with a large number of outcomes of the random parameters [2]. However, the difficulties we are facing are much more challenging than this, as our program is a stochastic program with integrality constraints in the last stage. Adding integrality restrictions to the last stage problem significantly increases the complexity of the problem. The main difficulty in solving stochastic integer programs is that, when integrality restrictions are present in the last stage, the last stage value function is not necessarily convex about previous stage variables but only lower semi-continuous. Thus, most standard decomposition approaches that work by building a precious convex under-estimator of the value function break down when last stage integer variables are present.

In conclusion, the stochastic model arising from the TpTRP is hard to solve as an mixed-integer multi-stage stochastic program.

3 Simplifications

Instead of focusing on the complex stochastic programming model and trying to solve it, in this work we propose to solve the TpTRP by Stochastic Dynamic Programming (SDP) using a discretization of the state space. Due to the discretization, we will only solve an approximation of the original TpTRP. The purpose of this section is to derive bounds on the cost of the optimal routing policy, which can be used to evaluate the quality of the solution obtained by SDP. To obtain a lower bound we derive an analytic solution of the deterministic version of the TpTRP, where we assume that we know all future traffic volumes ahead of making the routing decisions. To obtain upper bounds we analyse several plausible (although not optimal) routing policies of increasing complexity.

3.1 Deterministic problem

Firstly, we consider the deterministic version of the TpTRP in which all the volumes of traffic $T^\tau, \forall \tau \in \Gamma$ are known in advance. Without loss of generality, we can reorder these traffics into a non-increasing order $T^1 \geq T^2 \geq ... \geq T^{|\Gamma|}$. The following lemma developed by Matthieu Chardy [1] (in an unpublished technical report) gives the necessary conditions that the optimal routing policy should satisfy.

Lemma 3.1. *In the optimal routing policy for the deterministic case, the θ-th highest volume of traffic for all network providers apart from the cheapest one is zero, namely the ISP will incur a charge only from the cheapest network provider.*

Proof. We proof this lemma by contradiction.

Without loss of generality, we assume that $c_i > c_{i_{min}}, \forall i \neq i_{min}$. For a specific routing policy, let $u_i, i \in I$ be the volume of the θ-th highest volume of traffic shipped by network provider i. So that the cost charged on the ISP by this routing policy is $\sum_{i \in I} c_i u_i$.

Assume that there is an $i_0 \neq i_{min}$ and $u_{i_0} > 0$, we show that we can reduce the objective value (cost) with a reallocation of the traffics:

- reallocate u_{i_0} and all the lower traffics which were previously sent by network provider i_0 to network provider i_{min};
- leave all the rest allocation intact.

[1] Orange Labs, 38-40 rue du général Leclerc, BP 92130, Issy-les-Moulineaux.

With this reallocation, on one hand $u_{i_0}^{new} = 0$, that is we reduce the cost by $c_{i_0}u_{i_0}$. On the other hand, since all the reallocated traffics are no greater than u_{i_0}, therefore $u_{i_{min}}^{new} \leq c_{i_{min}}(u_{i_0} + u_{i_{min}})$, that is at worst the objective increases by $c_{i_{min}}u_{i_0}$. The total change in cost is thus $(c_{i_{min}} - c_{i_0})u_{i_0}$, which is strictly negative since $c_{i_0} > c_{min}$. So, we have a better solution from the reallocation, and proved by contradiction that all the network providers apart from the cheapest one get a top-percentile traffic of zero.

□

From this lemma we can derive the optimal routing policy for the deterministic case:

- No traffic is split, i.e. every time interval's traffic is shipped by a single network provider.
- The $n(\theta-1)$ highest traffics are allocated equally among the n network providers.
- All the other lower volume traffics (as many as $|\Gamma| - n(\theta - 1)$) are allocated to the cheapest network provider.

By implementing this optimal routing strategy, all network providers apart from the cheapest one get $(\theta-1)$ periods of traffic, therefore the charge to the ISP is zero as the θ-th highest volume of traffic is left to be zero. On the other hand, the top θth time interval for the cheapest network provider has a volume of $T^{n(\theta-1)+1}$, which introduces a cost of $c_{i_{min}}T^{n(\theta-1)+1}$. As a result, the total cost charged to the ISP by involving the optimal routing policy is equal to $c_{i_{min}}T^{n(\theta-1)+1}$.

As we assume that we have full knowledge of the traffic ahead in time, the optimal routing policy in the deterministic situation is not implementable. However, it provides us with lower bound on all the stochastic routing policies. This lower bound can be used in practise to evaluate how good a routing policy is.

3.2 Implementable routing policies

The aim of this section is to present three plausible routing policies of varying complexity for which we can analytically derive the expected cost. Throughout this section we need to assume:

(A1) All traffics $T^\tau, \tau \in \Gamma$ are independent and identically-distributed and have compool support: $[T_l, T_u]$.

(A2) We do not consider splitting the traffic, i.e. the whole traffic T^τ for every time interval τ is routed to a single network provider.

Let $T^\tau, \tau \in \Gamma$ be i.i.d random variables with probability distribution function $F(x) = pr(T^\tau \leq x), x \in [T_l, T_u]$.

We start by deriving an expression for the expectation of the top-θth volume of traffic. Let $Y^\theta_{|\Gamma|}$ be the the volume of the θ-th highest out of $|\Gamma|$ traffics. The probability distribution function of $Y^\theta_{|\Gamma|}$ is:

$$F_{Y^\theta_{|\Gamma|}}(x) = Pr(Y^\theta_{|\Gamma|} \leq x) = \sum_{0 \leq j \leq \theta} B(j||\Gamma|, 1 - F(x)),$$

where $B(j||\Gamma|, 1-F(x))$ is the probability mass function of the binomial distribution, which represents the probability that exactly j out of $|\Gamma|$ traffics are greater than x. As $F_{Y^\theta_{|\Gamma|}}(x)$ is a absolutely continuous distribution function, we define its density function as $f_{Y^\theta_{|\Gamma|}}(x)$ which satisfies $\int_{T_l}^{T_u} f_{Y^\theta_{|\Gamma|}}(x)dx = 1$ and $F_{Y^\theta_{|\Gamma|}}(x) = \int_{T_l}^{x} f_{Y^\theta_{|\Gamma|}}(t)dt, \forall x \in [T_l, T_u]$. Accordingly, the expectation of the volume of the top-θth traffic is given by:

$$E[Y^\theta_{|\Gamma|}] = \int_{T_l}^{T_u} x f_{Y^\theta_{|\Gamma|}}(x)d(x).$$

Single-homing Routing Policy (SRP)

The first routing policy we consider is single-homing, e.g. send everything by the cheapest network provider. The expected cost charged on the ISP will be:

$$Cost_{\mathbf{SRP}} = c_{i_{min}} E[Y^\theta_{|\Gamma|}] = c_{i_{min}} \int_{T_l}^{T_u} x d(\sum_{0 \leq j \leq \theta} B(j||\Gamma|, 1 - F(x))) \quad (3.1)$$

Trivial Multi-homing Routing Policy (TMRP)

Clearly the SRP is not good since we waste the $(\theta - 1)$ 'free' time intervals on all network providers apart from the cheapest one. Instead we consider the 'trivial' multi-homing routing policy that randomly routes $(\theta - 1)$ time intervals' traffic to every network provider and the rest to the cheapest one. In this way the ISP is only charged by the cheapest network provider, but uses the free time intervals of all network providers. The expected cost is accordingly:

$$Cost_{\mathbf{TMRP}} = c_{i_{min}} E[Y^{\theta}_{|\Gamma|-(n-1)(\theta-1)}]$$

$$= c_{i_{min}} \int_{T_l}^{T_u} x d(\sum_{0 \le j \le \theta} B(j||\Gamma| - (n-1)(\theta-1), 1 - F(x))) \quad (3.2)$$

Analytical Routing Policy (ARP)

It is trivial that the expected cost of TMRP is always lower than SRP. This means the routing policy derived from the single-homing architecture performs worse than the Trivial Multi-homing Routing Policy. Now the question is, whether we can improve on the TMRP.

The TMRP sends all the remaining time intervals' traffic to the cheapest network provider after filling the free time intervals of every network provider. However, it is not obvious that the lowest expected future cost is incurred by choosing the cheapest network provider to send all the remaining traffics. As an alternative we consider sending all the remaining traffics to the network provider with the least expected future cost. Here we focus on the minimum actual cost we are going to pay instead of simply choosing the network provider with the least per unit cost. Following we give the Analytical Routing Policy (ARP) in detail:

- Firstly, fill all the network providers' free $(\theta - 1)$ time intervals with the first $n(\theta - 1)$ traffics;
- Secondly, make decision on where to direct the $(n(\theta-1)+1)$st, and all the following time interval's traffic by comparing the future expected cost after time interval $n(\theta - 1)$. This means, we choose the decision which involves the least expected cost from n potential decisions: $D = \{d_1, d_2, ..., d_n\}$, where d_i represents 'allocate all the remaining traffics to network provider i. The expected cost implied by decision $d = d_i$ is:

  $$Cost_{d=d_i} = c_i E[\theta\text{th highest out of } \hat{T}_i^1, ..., \hat{T}_i^{\theta-1}, T^{n(\theta-1)+1}, ..., T^{|\Gamma|}],$$

  where $\hat{T}_i^j$ (known value) is the j-th highest volume of traffic that already been sent via network i, while $T^{n(\theta-1)+1}, ..., T^{|\Gamma|}$ are not yet revealed traffics. Comparing the expected cost implied by every decision we get the routing policy and thus the future expected cost:

  $$Cost_{\mathbf{ARP}} = \min_{d \in D} Cost_{d=d_i}. \quad (3.3)$$

3.3 An example

In this section we give an example of TpTRP in which all time intervals' traffic follow a same uniform distribution and derive the expected costs of implementing the above routing policies on it. They will serve as benchmarks to judge the quality of the routing policy obtained by SDP.

Assume that we have 2 network providers with cost $c_1 < c_2$. Both of them divide the charging period into 10 time intervals and charge based on the 3rd highest volume of traffic. Assume further that all time intervals' traffic follow a same uniform distribution with a cumulative distribution function:

$$F_U(x) = \begin{cases} 0, & 0 \le x \le 6000 \\ \frac{x-6000}{8000}, & 6000 < x < 14000 \\ 1, & x \ge 14000 \end{cases}$$

Let Y_m^k be the k-th highest traffic volume out of m random traffic. Then the distribution function of Y_m^k is

$$F_{Y_m^k}(x) = Pr(Y_m^k \le x)$$

$$= \sum_{0 \le j < k} B(j|m, 1-F_U(x)) = \sum_{0 \le j \le k} \left(\begin{array}{c} m \\ j \end{array} \right) (\frac{x-6000}{14000-6000})^{m-j} (\frac{14000-x}{14000-6000})^j.$$

And its expectation is

$$E[Y_m^k] = \int_{6000}^{14000} x d(F_{Y_m^k}(x)) = 14000 - \frac{k(14000-6000)}{m+1}.$$

According to formulae (3.1) and (3.2), the expected cost of implementing the SRP and TMRP are thus:

$$Cost_{\mathbf{SRP}} = c_1 E[Y_{10}^3] = 10 * (14000 - \frac{3 * (14000-6000)}{10+1}) = 118181.82.$$

$$Cost_{\mathbf{TMRP}} = c_1 E[Y_8^3] = 10 * (14000 - \frac{3 * (14000-6000)}{8+1}) = 113333.33.$$

Similarly, we can compute the expected cost of applying the Deterministic Routing Policy (DRP) if assuming we know all the traffic volumes in advance. This gives

$$Cost_{\mathbf{DRP}} = c_1 E[Y_{10}^5] = 10 * (14000 - \frac{5 * (14000-6000)}{10+1}) = 103636.36.$$

Now we show how to implement the ARP on the same example. It consists of two steps: Firstly allocate T^1, T^2 to network provider 1 and T^3, T^4 to network provider 2. Secondly, compare the expected cost of the following two choices to make the routing decision for all following time stages (here ω_τ represent the realisation of traffic T_τ):

1. send $T^5, ..., T^{10}$ to network 1

   $Cost_{d=d_1} = c_1 E[\text{top-3rd out of } \omega_1, \omega_2, T^5, ..., T^{10}]$,

2. send $T^5, ..., T^{10}$ to network 2

   $Cost_{d=d_2} = c_2 E[\text{top-3rd out of } \omega_3, \omega_4, T^5, ..., T^{10}]$,

Take $d = d_1$ for example, let A be the random variable representing the 3rd highest volume of traffic out of $\omega_1, \omega_2, T^5, ..., T^{10}$, and $F_A(x)$ be the probabilistic distribution function of A. Assume $\omega_1 \geq \omega_2$, then

$$F_A(x) = Pr(A \leq x) = \begin{cases} F_6^0(x), & \text{if } \omega_1 \geq x, \omega_2 \geq x \\ F_6^1(x), & \text{if } \omega_1 \geq x, \omega_2 \leq x \\ F_6^2(x), & \text{if } \omega_1 \leq x, \omega_2 \leq x \end{cases}$$

So that $Cost_{d=d_1} = c_1 E[A] = \int_{6000}^{14000} x d(F_A(x))$.

For example, if $\omega^1 = 13690, \omega^2 = 13361$ and $\omega^3 = 7730, \omega^4 = 7238$, we can compute

$$Cost_{d=d_1} = c_1 E[\text{top-3rd out of } 13690, 13361, T^5, ..., T^{10}] = 127659,$$

$$Cost_{d=d_2} = c_2 E[\text{top-3rd out of } 7730, 7238, T^5, ..., T^{10}] = 127295.$$

Comparing this two expected cost we find the decision $d = 2$ is better and send all the remaining traffic to network provider 2. In fact with provider costs $c1 = 10$ and $c2 = 12$, when ω_1, ω_2 are high enough (greater than 13000) and ω_3, ω_4 are relatively low (say less than 8000), it might be better to allocate all the remaining traffic to the expensive network provider.

In theory, the ARP improves the TMRP by comparing the expected future cost to make routing decision thus should cost less than the TMRP. However, we cannot evaluate the cost of the ARP analytically. In Section 5.2 we evaluate the ARP on $1,000,000$ random traffic scenarios and find

$$Cost_{\mathbf{ARP}} = 113351.84 \pm 12.08.$$

4 Stochastic dynamic programming model

We have seen in Section 2.3 that the stochastic programming model of the TpTRP is computationally intractable. On the other hand, Section 3 has presented two radical simplifications of the decision process that result in analytically solvable models (the deterministic model and the ARP model). In this section we show how, by a discretization of the state space, the TpTRP can be solved by Stochastic Dynamic Programming (SDP). We now define the main modelling elements we need in the SDP model.

4.1 Main modelling elements

Stages

Stages in this problem can be defined naturally by the time interval. At the beginning of every time stage, we observe on the current state and make routing decisions for this time interval's traffic.

States

In our problem, at the beginning of time interval τ, we know all the previous realisations of traffic volumes $\hat{T}^t, t = 1, ..., \tau - 1$ and routing decisions $x_i^t, t = 1, ..., \tau - 1$. The implied usage $\hat{T}_i^t = \hat{T}^t x_i^t, t = 1, ..., \tau$ of network i can be computed. These define the current state S^τ of the system. However in SDP, we aim to build up value functions $V_\tau(S^\tau)$ which represents the minimum expected cost from time interval τ to the end, where the current state is S^τ. In our case the cost is solely determined by the θ-th highest volume of traffic for every network provider, at the end of the charging period. Since traffic volumes $\hat{T}_t^{j,\tau}, j > \theta, \tau = 1, ..., |\Gamma|$ cannot impact on the cost, we delete them from the state space. Thus the state variable is described by

$$S^\tau = (\hat{T}_1^{1,\tau}, \hat{T}_1^{2,\tau}, ..., \hat{T}_1^{\theta,\tau}; \hat{T}_2^{1,\tau}, \hat{T}_2^{2,\tau}, ..., \hat{T}_2^{\theta,\tau}; ...; \hat{T}_n^{1,\tau}, \hat{T}_n^{2,\tau}, ..., \hat{T}_n^{\theta,\tau}).$$

In addition, as SDP finds a representation of $V_\tau(S^\tau)$ by tabulating its value for every possible state S^τ. In order to obtain a computationally tractable model we need to discretize the state space.

Decisions

So far we have assumed that the traffic of any time interval is routed by a single network provider. In practice, it is possible for us to split any time

interval's traffic and send it through more than one network. Under top-percentile pricing network provider i computes the cost based on the θ-th highest volume of shipped traffic, namely $cost_i = f(\hat{T}_i^{\theta})$. If at the beginning of time interval τ the θ-th highest volume of traffic sent via network provider i is $\hat{T}_i^{\theta,\tau}$, any allocation of less than $\hat{T}_i^{\theta,\tau}$ in the following time interval will not change the price charged by this network provider. Therefore, the ISP can ship at most $\sum_{i\in I} \hat{T}_i^{\theta,\tau}$ units of data without increasing the cost. We call this amount, the 'Free' volume of traffic in the following parts of this paper.

$$\hat{T}_{Free}^{\tau} = \sum_{i\in I} \hat{T}_i^{\theta,\tau}, \qquad (4.1)$$

We can ship up to this amount of traffic using this 'free capacity' of each network provider and make the routing decision only on the traffic in excess of this amount. We define this amount of traffic as the 'additional traffic':

$$T_{Add}(S^{\tau}) = \begin{cases} T^{\tau} - \hat{T}_{Free}^{\tau} & \text{if} \quad T^{\tau} - \hat{T}_{Free}^{\tau} > 0 \\ 0 & \text{otherwise} \end{cases}$$

Given a state S^{τ}, we make our decision on where to direct the 'additional traffic' to minimize the value function $V_{\tau}(S^{\tau})$. Furthermore for simplification, in this model we do not split the 'additional traffic'. Namely at every stage, we have a decision set with only n choices: $X = \{d_1, ..., d_n\}$, where $d_i, i \in I$ represents routing the whole 'additional traffic' via network provider i. According to the discussion in Section 2.1, making decision on additional traffic is implementable.

Formulations

An SDP problem can be written in terms of recursion that relates the value of being in a particular state at one point in time to the value of the states that we are carried into at the next point in time. The Bellman's equation for deterministic problems is: [8]

$$V_{\tau}(S^{\tau}) = \min_{x_{\tau}\in X}(C_{\tau}(S^{\tau}, x_{\tau}) + V_{\tau+1}(S^{\tau+1}(S^{\tau}, x_{\tau}))),$$

where $C_{\tau}(S^{\tau}, x_{\tau})$ is the immediate cost of taking action x_{τ} at state S^{τ}, $S^{\tau+1}$ is the state we transition to if we are currently in state S^{τ} and take action x_{τ}.

In the stochastic case the Bellman's equation becomes:

$$V_{\tau}(S^{\tau}) = \min_{x_{\tau}\in X}(\sum_{\omega_{\tau}\in\Omega^{\tau}} pr(\omega_{\tau}) V_{\tau+1}(S^{\tau+1}(S^{\tau}, x_{\tau}, T^{\tau}(\omega_{\tau})))), \qquad (4.2)$$

where Ω^τ is the set of all the potential realisations of the new traffic T^τ and $pr(\omega_\tau)$ is the probability that $T^\tau = \omega_\tau$ occurs. Now $S^{\tau+1}$ is a function of the previous state S^τ, the decision x_τ and the realization of traffic T^τ.

The update from $(S^\tau, x_\tau, T^\tau(\omega_\tau))$ to $S^{\tau+1}$ is as follows (assuming the decision is $x_\tau = d_{\hat{i}}$):

1. If the new traffic $T^\tau(\omega_\tau)$ is less than or equal to T^τ_{Free} (as in formula 4.1):

$$T^{j,\tau+1}_{\hat{i}} = T^{j,\tau}_{\hat{i}}, j = 1, ..., \theta$$
$$T^{j,\tau+1}_{i} = T^{j,\tau}_{i}, j = 1, ..., \theta; i \neq \hat{i}.$$

2. If the new traffic $T^\tau(\omega_\tau)$ is greater than T^τ_{Free}:

$$\tilde{T}^{j,\tau+1}_{\hat{i}} = \tilde{T}^{j,\tau}_{\hat{i}}, j = 1, ..., \theta - 1$$
$$\tilde{T}^{\theta,\tau+1}_{\hat{i}} = T^\tau(\omega_\tau) - \sum_{i \neq \hat{i}} T^{\theta,\tau}_i,$$

   then reorder these θ values into a non-increasing order to obtain $(T^{1,\tau+1}_{\hat{i}}, T^{2,\tau+1}_{\hat{i}}, ..., T^{\theta,\tau+1}_{\hat{i}})$. For network provider $i \neq \hat{i}$, the state variable is remain the same:

$$T^{j,\tau+1}_{i} = T^{j,\tau}_{i}, j = 1, ..., \theta; i \neq \hat{i}.$$

In addition, at the last stage $|\Gamma| + 1$, we know all the routing decisions and outcomes of traffic, we compute the cost charged on the ISP as follows:

$$V_{|\Gamma|+1}(S^{|\Gamma|+1}) = V_{|\Gamma|+1}(T^{1,|\Gamma|}_1, T^{2,|\Gamma|}_1, ..., T^{\theta,|\Gamma|}_1; ...; T^{1,|\Gamma|}_n, T^{2,|\Gamma|}_n, ..., T^{\theta,|\Gamma|}_n) = \sum_{i \in I} c_i T^{\theta,|\Gamma|}_i.$$

So that the complete recurrence function is:

$$\begin{cases} V_\tau(S^\tau) = \min\limits_{x_\tau \in X}(\sum\limits_{\omega_\tau \in \Omega^\tau} pr(\omega_\tau) V_{\tau+1}(S^{\tau+1})), \tau = 1, ..., |\Gamma| \\ V_{|\Gamma|+1}(S^{|\Gamma|+1}) = \sum\limits_{i \in I} c_i T^{\theta,|\Gamma|}_i \end{cases}$$

5 Numerical Results

5.1 Test Problems

In this section we give some numerical results on several small instances of the TpTRP. For clarity, we firstly characterise and index these instances which are examined in the later part of this section.

Instance	Parameters			Stochastic Information	
	$\|\Gamma\|$	θ	n	distribution	time dependency
Instance 1	5	2	2	$U(6000, 12000)$	i.i.d.
Instance 2	10	3	2	$U(6000, 14000)$	i.i.d.
Instance 3	10	3	2	uniform	see Fig. 5
Instance 4	10	3	2	truncated $N(10000, 10^6)$	i.i.d.
Instance 5	10	3	2	truncated normal	see Fig. 6

Table 1: List of TpTRP Instances

Table 1 summarises the instances used. Instance 1 is the one that we failed to solve in the stochastic programming section, which is also the smallest one listed here. For the other instances, we assume that we divide the modelling region into 10 time intervals and cost are based on the time interval with the $\theta = \lfloor q * |\Gamma| \rfloor = 3$rd ($q = 0.3$) highest volume of traffic. In all cases we use 2 network providers ($n = 2$) with costs $c_1 = 10, c_2 = 11, 12$ or 15. Apart from Instance 1, the other instances differ by the assumptions made on the random traffic. In instance 2 and 4 the traffic in every period follows the same uniform ($U(6000, 14000)$ in Instance 3) or normal ($N(10000, 10^6)$ in Instance 4) distribution. Instance 3 and 5 on the other hand, use traffic distributed according to a time varying uniform or normal distribution. The parameter for each time interval are displayed in Figures 4 and 5. Note that Instance 4 and 5 uses a truncated normal distribution in which traffic outside the 99.7% ($\pm 3\sigma$) confidence region is projected onto the boundary of the region.

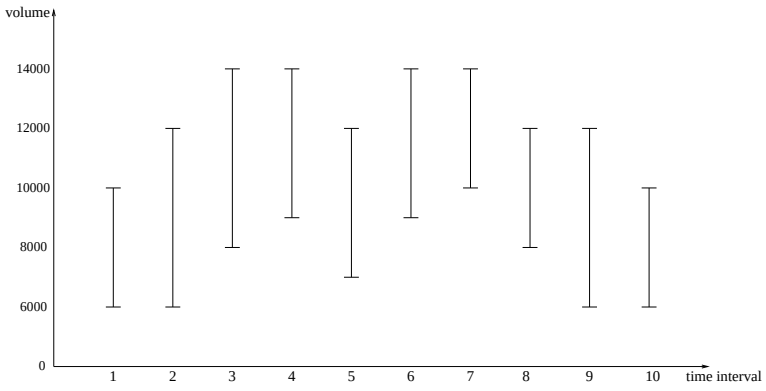

Figure 4: Upper and lower bounds for uniform distributions in Instance 3

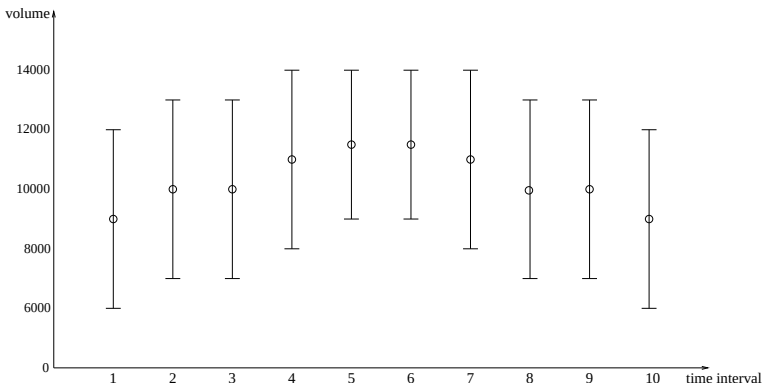

Figure 5: Mean and 99.7% confidence region for normal distributions in Instance 5

5.2 Bounds on the TpTRP optimal solution for i.i.d. traffic

In section 3.2, we have presented several simple routing policies of TpTRP and also the theoretical expected cost of applying them on an example with uniform i.i.d. traffic (same as Instance 2). In this section however, we will test these plausible routing policies on a group of $1,000,000$ random scenarios, give the mean costs of implementing them in practice. As mentioned in Section 3.2, the ARP is only applicable in case where traffic for different time intervals are independent and identically distributed. Therefore we consider Instance 2 and 4 and summarise the numerical test result in Table 2. The entries in this table is given in a format of '**Mean Cost±Standard Deviation**'.

Instance	c_2	**SRP**	**TMRP**	**ARP**	**DRP**
Instance 2	11	118193.04±10.32	113352.46±12.07	113138.24±12.27	103659.85±11.54
	12	118193.04±10.32	113352.46±12.07	113351.84±12.08	103659.85±11.54
	15	118193.04±10.32	113352.46±12.07	113352.46±12.07	103659.85±11.54
Instance 4	11	129490.72±13.99	122749.70±20.28	122645.30±20.72	107256.27±26.01
	12	129490.72±13.99	122749.70±20.28	122749.70±20.28	107256.27±26.01
	15	129490.72±13.99	122749.70±20.28	122749.70±20.28	107256.27±26.01

Table 2: Numerical result (mean cost and standard deviation) of implementing simple routing policies on $1,000,000$ scenarios

We considered three different choices of the per unit cost for network provider 2 ($c_2 = 11, 12$ or 15), where $c_1 = 10$. From this table we can see, the inequality SRP $\geq$ TMRP $\geq$ ARP $\geq$ DRP holds for all choices of c_2. Apart from the ARP, the mean cost of all the other routing policies does not vary with the change of c_2, because they use no more than the free time intervals of network provider 2. Moreover, in case where $c_2 = 11$, the ARP performs

definitely better than the TMRP by using network provider 2 to send several scenarios' remaining traffic. For example for Instance 2 ($c_2 = 11$), there are $75,932$ out of $1,000,000$ scenarios using network provider 2 to send the remaining traffics, which leads to a definitely lower mean cost for the ARP than TMRP. However, in case where $c_2 = 12$ and $c_2 = 15$, the difference between these two routing policies is not significant as the standard deviation is greater than the difference in mean cost.

In addition, comparing the mean cost for Instance 2 given in Table 2 with the theoretical expected cost of applying those simple routing policies (given in section 3.3), we see that in all cases, the theoretically obtained value is within $\pm 2\sigma$ of the result obtained by simulation, confirming our analysis.

5.3 SDP routing policy

To obtain the optimal routing policy by SDP, we discretize the possible traffic volumes into a specific number of levels with equal size, and approximate the continuous distribution functions of random traffic by discrete ones. For example, if we discretize the traffic region $[0, 14000]$ equally into 14 levels, then we use $\Omega = \{6000, 7000, ..., 13000\}$ as the set of potential realizations of traffic. In case of Instance 2 where uniform distribution is considered, we approximate it by the discrete distribution $pr(1000 * i) = 1/8, i \in \{6, 7, ..., 13\}$. Namely we always round the traffic down to the whole thousand.

With the discretization, we can solve the resulting TpTRP with SDP techniques as shown in Section 4. By evaluating the minimization problem (4.2), we get a decision on where to route the additional traffic at every time interval, proceeding backwards in time. Finally we end up with a routing table that gives for every stage and every state (i.e. the θ highest traffic volumes by every network provider), the optimal routing decision for this period's additional traffic.

To evaluate the quality of this routing policy, we examine it in a simulation of $1,000,000$ random scenarios taken from the (original) continuous distribution. We obtain a routing decision by rounding the current (continuous) state down to the nearest tabulated discrete state.

Mean cost

For every problem instance, we experiment with different discretization levels, to see how the discretization reflects the optimal routing policy. We also give results of the numerical test on SRP, TMRP and DRP as a comparison.

Looking at Table 3 we can see, the SDP routing policy performs much better (discretizing into more than 7 levels) than SRP and TMRP, and no

Instance	SRP	TMRP	DRP	levels	SDPRP
Instance 2	118193.04±10.32	113352.46±12.07	103659.85±11.54	7	107811.61±12.90
				14	106602.73±12.33
				28	106203.10±12.17
Instance 3	114351.48±7.61	104311.21±7.89	102319.77±7.05	7	105256.80±8.44
				14	103853.99±7.71
				28	103375.22±7.54
Instance 4	129490.72±13.99	122749.70±20.28	107256.27±26.01	7	115349.65±27.07
				14	113684.41±25.42
				28	112426.28±24.37
Instance 5	134311.75±13.88	123051.54±19.42	110998.99±22.17	7	117998.14±23.24
				14	116955.95±22.35
				7	116394.02±22.00

Table 3: Numerical result (mean cost and standard deviation) of implementing SDP routing policy (SDPRP) on $1,000,000$ scenarios with $c_2 = 12$

more than 10% higher than the lower bound – the DRP. Numerical results in Table 3 show that the average cost is getting lower as the discretization is getting finer. This means firstly, the optimal routing policy is changing as the number of level changes, and secondly, the better results might be achievable from an even finer discretization. However, both computation time and memory use increase drastically with the use of more discretization levels, therefore a finer discretization is not computable.

We summarise the numerical test results for $c_2 = 11$ and $c_2 = 15$ in Table 4 and 5 respectively. From these tables we can see, they are almost of the same structure as when $c_2 = 12$. Therefore, we have reason to believe that our SDP model is a good approximation of the original TpTRP and provides us implementable routing policy which performs significantly better than the trivial routing policies.

Instance	SRP	TMRP	DRP	levels	SDPRP
Instance 2	118193.04±10.32	113352.46±12.07	103659.85±11.54	7	107219.29±12.69
				14	106090.86±12.18
				28	105734.36±12.03
Instance 3	114351.48±7.61	104311.21±7.89	102319.77±7.05	7	104728.46±8.15
				14	103543.13±7.53
				28	103140.84±7.42
Instance 4	129490.72±13.99	122749.70±20.28	107256.27±26.01	7	112736.85±27.45
				14	112284.40±25.95
				28	111243.77±25.93
Instance 5	134311.75±13.88	123051.54±19.42	110998.99±22.17	7	116876.23±23.50
				14	116048.43±22.47
				7	115527.05±22.15

Table 4: Numerical result (mean cost and standard deviation) of implementing SDPRP on $1,000,000$ scenarios with $c_2 = 11$

Instance	SRP	TMRP	DRP	levels	SDPRP
Instance 2	118193.04±10.32	113352.46±12.07	103659.85±11.54	7	109022.21±13.42
				14	107432.60±12.60
				28	106932.48±12.38
Instance 3	114351.48±7.61	104311.21±7.89	102319.77±7.05	7	106260.86±9.04
				14	104197.98±7.92
				28	103679.88±7.67
Instance 4	129490.72±13.99	122749.70±20.28	107256.27±26.01	7	121318.02±23.87
				14	118532.91±22.55
				28	116904.49±22.72
Instance 5	134311.75±13.88	123051.54±19.42	110998.99±22.17	7	119729.13±22.76
				14	118916.88±21.67
				7	118294.11±21.38

Table 5: Numerical result (mean cost and standard deviation) of implementing SDPRP on $1,000,000$ scenarios with $c_2 = 15$

Resource consumption

In our state variable

$$S^\tau = (\hat{T}_1^{1,\tau}, \hat{T}_1^{2,\tau}, ..., \hat{T}_1^{\theta,\tau}; \hat{T}_2^{1,\tau}, \hat{T}_2^{2,\tau}, ..., \hat{T}_2^{\theta,\tau}; ...; \hat{T}_n^{1,\tau}, \hat{T}_n^{2,\tau}, ..., \hat{T}_n^{\theta,\tau}),$$

parameters are the θ highest volume of traffics for every network provider. Thus we have naturally $\hat{T}_i^{1,\tau} \geq \hat{T}_i^{2,\tau} \geq ... \geq \hat{T}_i^{\theta,\tau}, \forall i \in I$. If the traffic region ($[0, 14000]$ in our instances) is divided into L levels, there are at most $C_{L+\theta-1}^{\theta} = \begin{pmatrix} L+\theta-1 \\ \theta \end{pmatrix}$ [2] underlying outcomes of the state variable $S_i^\tau = (\hat{T}_i^{1,\tau}, \hat{T}_i^{2,\tau}, ..., \hat{T}_i^{\theta,\tau})$. In case where there are n network providers and $|\Gamma|$ time intervals in the TpTRP instance, we have a state space with $|\Gamma|(C_{L+\theta-1}^{\theta})^n$ states in the underlying SDP model.

We need to compare the expected future cost $V_{\tau+1}(S^{\tau+1})$ for all states in the subsequent stage $\tau + 1$. After all routing decisions for stage τ have been computed, the costs $V_{\tau+1}(S^{\tau+1})$ are not required anymore. Therefore it is sufficient to store the $V_\tau(S^\tau)$ value for two subsequent stages τ and $\tau+1$. For the routing decision x_τ we have two choices. In the simplest case, all routing decisions are computed (and stored) ahead of time. Implementing a routing policy then amounts to simply looking up the corresponding x in a routing table. However, it calls for a great quantity of computer memory to keep the big decision matrix. For example in Instance 2 with 28 levels, the decision matrix has $10(C_{30}^3)^2 = 164836000$ entries, it needs at least 628.80Mb to store the decision matrix only, and at least 251.52M to store the value $V_\tau(S^\tau)$ for two stages.

[2] $\begin{pmatrix} L+\theta-1 \\ \theta \end{pmatrix}$ is the number of possibilities which satisfies $\hat{T}_i^{j,\tau} \in \Omega^\tau, j = 1, ..., \theta$ and $\hat{T}_i^{1,\tau} \geq \hat{T}_i^{2,\tau} \geq ... \geq \hat{T}_i^{\theta,\tau}$.

Also, running time is another important issue that we need to be concerned about. In Table 6 we give a summary of the running time (to get the decision table only) and the memory consumption (theoretical) in solving the SDP model to optimality for Instance 2.

Levels	Problem Size		Resource Consumption	
	Level Length	No. of States	Running Time	Memory Consumption
7	2000	7056	0.194s	0.38Mb
14	1000	313600	15.628s	16.75Mb
28	500	16483600	11487.095s	880.32Mb

Table 6: Comparison of problem size and resource consumption - original model for Instance 2

From Table 6 we can see that the number of state variables and thus the memory consumption grows sharply with the number of discretization levels of the traffic region. Namely, as quality of the routing policy enhances with the increasing of number of levels, we have to pay for this enhancement.

Alternatively, apart from storing all the decisions in a big matrix, we can regenerate the routing decisions x_τ for a given stage only when needed. As the SDP model is solved from back to the top, keep one decision means we choose to drop all the decisions we have made for time stages $\{\tau+1, ..., |\Gamma|\}$, which would be needed in the later stages. Hence to implement a routing policy requires running the SDP model for $|\Gamma|$ times. This costs us a great deal of time. Table 7 gives a summary of running time and memory consumption in this case. Note that as we need to know the current state to decide which decision to generate at every time stage, we follow a single scenario in this experiment. The running time shown in the table consists both the decision generating time and the implementing time on this scenario. (Although comparing with generating the decision, the implementation cost little.)

Levels	Problem Size		Resource Consumption	
	Level Length	No. of States	Running Time	Memory Consumption
7	2000	7056	0.676s	0.11M
14	1000	313600	117.937s	4.79M
28	500	16483600	37687.800s	251.52M

Table 7: Comparison of problem size and resource consumption - alternative model for Instance 2

Looking at Table 7 we can see, although the memory consumption reduced to about 1/4 of the previous amount, we have to spend much more time on getting the decision for a single run. Moreover, this alternative model maybe challenging to implement in reality. As shown in Figure 3, we know the cumulative value of $T^{\tau-1}$ at the end of time stage $\tau-1$, where we can run the SDP model to get the routing decision for time stage τ. However, the end of time stage $\tau-1$ is also the beginning of time stage τ, where we need

to implement the decision x^τ at once. So that we have no time to wait for the decision x^τ been generated, although it needs a significant amount of time to do.

In general, there is a trade-off between running time and computer memory consumption. No matter which of these two is the critical issue, it does prevent the use of the SDP model for finer discretization levels or larger problem instances. This is the well known 'curse of dimensionality' in dynamic programming. In conclusion, the SDP model provide us a routing policy which is significantly superior to the simple routing policies and close to the lower bound (optimal cost in deterministic case). However, the huge number of states prevents the use of the SDP model on larger problem instances.

5.4 Implementable decision rule by classification

As shown above, the optimal routing policy given by the SDP model is represented by a large discrete look-up table (7-dimensional for Instance 2, 3 and 4). This is caused by the fact that, in order to impose the traditional SDP technique on the TpTRP, we discretize the continuous traffic space into a discrete one. The discretization brings not only some errors within the discretization step, but also a large decision table which consumes a great amount of computer memory.

Realistic problems have much larger sizes than the ones we considered. For example the ISP divides a month into $10mins$ intervals which leads to 4320 stages. Clearly the model as stated is not applicable to problems of this size. Instead, we use a classification method (such as support vector machines [10]) to help identifying the conditions under which the more expensive provider (provider 2) should be used from the routing table. We then extrapolate these conditions to the large problem instance.

After investigating the discrete decision table for Instance 2 with 7 levels, we find that generally the SDP routing decisions can be simplified in the following way (assuming we have 2 network providers with $c_1 = 10$ and $c_2 = 12$):

- For time intervals $\{1, 2, 3\}$, send everything to the cheapest network provider (network provider 1);
- For time intervals $\{4, 5, 6, 7\}$, $(S^\tau = (\hat{T}_1^{1,\tau}, \hat{T}_1^{2,\tau}, \hat{T}_1^{3,\tau}; \hat{T}_2^{1,\tau}, \hat{T}_2^{2,\tau}, \hat{T}_2^{3,\tau}))$
    1. if $\hat{T}_1^{2,\tau} - \hat{T}_1^{3,\tau} \geq 1500$ and $\hat{T}_2^{1,\tau} \leq 2000$, send the additional traffic $T_{Add}(S_\tau)$ to network provider 2;
    2. otherwise send the additional traffic $T_{Add}(S_\tau)$ to network provider 1.

- For time intervals $\{8, 9, 10\}$,

    1. if $\hat{T}_2^{2,\tau} \leq 1500$, send the additional traffic $T_{Add}(S_\tau)$ to network provider 2;
    2. otherwise send the additional traffic $T_{Add}(S_\tau)$ to network provider 1.

This decision rule means, we prefer to send most time intervals' traffic by the cheapest network provider and no more than the additional amount of traffic by the second network provider. If the two 'free' (highest) traffics sent by the cheapest network provider are high enough (relatively higher than the 3rd highest one) and at the same time, the expected 3rd highest volume of traffic for network provider 2 is small enough (say no more than 1500 here), then sending the additional traffic to network provider 2 is better. Implementing this classified decision rule (CDR) on the instances, we get the following results:

Instance	SRP	TMRP	SDPRP - 7levels	CDR	DRP
Instance 2	118193.04±10.32	113352.46±12.07	107811.61±12.90	108805.82±13.03	103659.85±11.54
Instance 3	114351.48±7.61	104311.21±7.89	105256.80±8.44	108322.02±8.64	102319.77±7.05
Instance 4	129490.72±13.99	122749.70±20.28	115349.65±27.07	113195.86±25.77	107256.27±26.01
Instance 5	134311.75±13.88	123051.54±19.42	117998.14±23.24	120378.18±21.96	110998.99±22.17

Table 8: Numerical result (mean cost and standard deviation) of implementing CDR routing policy on $1,000,000$ scenarios with $c_2 = 12$

From this table we can see, the classified decision rule performs better than the TMRP in Instance 2, 4 and 5. Moreover in Instance 4, it seems even better than the exact SDP routing policy for in 7 levels model. Observing this, we can develop routing policy for instances with more time intervals. For example, in case where we extend the Instance 2 to 4320 time intervals and both network providers charge the ISP based on the 216th highest volume of traffic, implementing the following decision rule:

- For time intervals $\{1, 2, ..., 216\}$, send everything to the cheapest network provider (network provider 1);
- For time intervals $\{216, 217, ..., 4014\}$,

    1. if $\hat{T}_1^{215,\tau} - \hat{T}_1^{216,\tau} \geq 1500$ and $\hat{T}_2^{214,\tau} \leq 2000$, send the additional traffic $T_{Add}(S_\tau)$ to network provider 2;
    2. otherwise send the additional traffic $T_{Add}(S_\tau)$ to network provider 1.

- For time intervals $\{4015, 4016, ..., 4320\}$,

    1. if $\hat{T}_2^{215,\tau} \leq 1500$, send the additional traffic $T_{Add}(S_\tau)$ to network provider 2;
    2. otherwise send the additional traffic $T_{Add}(S_\tau)$ to network provider 1.

Test this routing decision on $1,000,000$ random scenarios (uniform distributed with mean 10,000), the result is shown in Table 9. It is clear that the extended routing policy performs better than the TMRP, although much higher than the lower bound.

Index	**SRP**	**TMRP**	**CDR**	**DRP**
mean cost ± s.d	136001.02 ±2.63	135999.15±2.64	135792.52±2.76	132020.52±3.67

Table 9: Numerical result (mean cost and standard deviation) of implementing CDR routing policy on $1,000,000$ scenarios with $c_2 = 12$

6 Conclusions and future works

The above experiments on the SDP routing policy shows that, the TpTRP can be solved by our SDP model for small instances. We demonstrate that a good routing policy can obtain significantly better results than any naive multi-homing routing policy. For instances with uniform or normal traffic distributions, our corresponding SDP model gives routing polices whose numerical result on random data is just slightly greater than the lower bound (given by DRP). However, for the real-world size problem (4320 time intervals and 5%-percentile pricing), modelling by the exact SDP yields too many states which consumes too much computer memory to run. As an alternation, we abstract a decision rule from the small size instances, which can be applied to real-world problems and the numerical test shows that it still outperforms the naive routing policies.

In conclusion, DP model is a promising model for small scale TpTRP. However, more work should be done in order to make the SDP model available for real-world sized problems. For example the Approximate Dynamic Programming [9] technique is a promising avenue to avoid the curse of dimensionality. We leave this as future work.

References

[1] E. Altman, T. Basar, T. Jimnez, and N. Shimkin, *Competitive routing in networks with polynomial costs*, IEEE Transactions on Automatic Control, 47 (2002), pp. 92–96.

[2] J. Birge and F. Louveaux, *Introduction to Stochastic Programming*, Springer, New York, 1997.

[3] M. Chardy, A. Ouorou, and T. VanDonselaar, *Optimization of interconnoction strategy in top-percentile pricing framework*, technical report, Orange Labs, France Telecom, 38-40 rue du général Leclerc, BP 92130, Issy-les-Moulineaux, 2009.

[4] D. Goldenberg, L. Qiu, H. Xie, Y. Yang, and Y. Zhang, *Optimizing cost and performance for multihoming*, ACM SIGCOMM Computer Communication Review, 34 (2004), pp. 79–92.

[5] M. Herzberg and F.Shleifer, *Optimization models for the design of bi-directional self-healing ring based networks*, Teletraffic Science and Engineering, 3 (1999), pp. 183–194.

[6] J. Levy, H. Levy, and Y. Kahana, *Top percentile network pricing and the economics of multi-homing*, Annals of Operations Research, 146 (2006), pp. 153–167.

[7] A. Odlyzko, *Internet pricing and the history of communications*, Computer Networks, 36 (2001), pp. 493–517.

[8] M. Paquet, A. Martel, and B. Montreuil, *A manufacturing network design model based on processor and worker capabilities*, International Journal of Production Research, 46 (2007), pp. 2009–2030.

[9] W. Powell, *Approximate Dynamic Programming - Solving the Curses of Dimensionality*, John Wiley & Suns, New Jersey, 2007.

[10] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, 1995.

[11] X. Wang and H. Schulzrinne, *Pricing network resources for adaptive applications*, IEEE/ACM Transactions on Networking (TON), 14 (2006), pp. 506–519.