Stochastic Shortest Path Problem with Delay Excess Penalty

Stefanie Kosuch [1]

Laboratoire de Recherche en Informatique, Université Paris Sud, Orsay, France

Abdel Lisser [2]

Laboratoire de Recherche en Informatique, Université Paris Sud, Orsay, France

Abstract

We study and solve a particular stochastic version of the Restricted Shortest Path Problem, the Stochastic Shortest Path Problem with Delay Excess Penalty. While arc costs are kept deterministic, arc delays are assumed to be normally distributed and a penalty per time unit occurs whenever the given delay constraint is not satisfied. The objective is to minimize the sum of path cost and total delay penalty.

Keywords: restricted shortest path, stochastic optimization, projected gradient algorithm, active set methods, branch-and-bound

1 Introduction

The Restricted Shortest Path Problem ($RSSP$) is a well studied extension of the famous Shortest Path Problem. It consists in finding a shortest or least

[1] Email: stefanie.kosuch@lri.fr
[2] Email: abdel.lisser@lri.fr

cost path from a node s to a node t in a graph subject to an additional constraint that typically models the restricted delay.
$RSSP$ has first been studied by Joksch in 1966 ([1]). Unlike the Shortest Path Problem that can be solved in polynomial time, $RSSP$ is part of the NP-complete problem class.
In this paper we study a particular stochastic version of $RSSP$, the Stochastic Shortest Path Problem with Delay Excess Penalty ($SSPD$). In this model the arc delays are random and we assume them to be independently normally distributed.
The problem has a simple recourse formulation. This means that the randomness of the delay is handled by introducing a penalty that occurs in case the delay constraint is not satisfied. The objective is to minimize the sum of path cost and expected total delay penalty.
This particular stochastic variant of $RSSP$ has been previously studied and shown to be NP-hard (see [3]). However, the authors of [3] assumed discrete distributions for the delay.
We propose to solve the problem by applying a problem specific branch-and-bound algorithm. The main idea of this algorithm is to search the admissible set (all directed paths from s to t) using a sort of depth-first search on the given directed graph. In order to reject subspaces of the search space, lower bounds are computed by solving the corresponding relaxed problems.

2 Problem Formulation

Let in the following $G = (V, A)$ be a directed, simple graph without directed cycles. To every arc $a \in A$ we assign a cost $c(a) > 0$ as well as a normally distributed delay with strictly positive mean represented by the random variable $\delta(a)$. We further assume for two distinct arcs a and a' that $\delta(a)$ and $\delta(a')$ are independent.
The *Stochastic Shortest Path Problem with Delay Excess Penalty (SSPD)* consists in finding a directed path between two given vertices s and t such that the cost of the path plus the expected delay cost are minimal. The delay cost is based on a penalty per time unit $d > 0$ that has to be paid whenever the total delay excesses a given maximum delay $D > 0$.

$SSPD$ can be formulated as a stochastic combinatorial optimization problem in the following way: Let $x \in \{0, 1\}^{|A|}$ such that each component x_a of x represents an arc $a \in A$. For a directed path P we define the corresponding $x = x(P)$ such that $x_a = 1$ if and only if $a \in P$. This leads to the following

mathematical formulation of the $SSPD$:

$$\min_{x\in\{0,1\}^{|A|}} \quad \mathbb{E}[j(x,\chi)] := \sum_{a\in A} c(a)x_a + d\cdot\mathbb{E}[[\sum_{a\in A}\delta(a)x_a - D]^+] \tag{1a}$$

$$\text{s.t. } \forall v\in V: \quad \sum_{\substack{w\in V:\\(v,w)\in A}} x_{(v,w)} - \sum_{\substack{w\in V:\\(w,v)\in A}} x_{(w,v)} = \begin{cases} 1 & \text{if} \quad v=s,\\ -1 & \text{if} \quad v=t,\\ 0 & \text{else.}\end{cases} \tag{1b}$$

Problem (1) can be compactly written as:

$$\text{(SSPD)} \quad \min_{x\in\{0,1\}^{|A|}} \quad \mathbb{E}[j(x,\chi)] \tag{2a}$$

$$\text{s.t. } Mx = b \tag{2b}$$

Remark that we can drop arbitrarily one of the constraints in (1b) in order to obtain a matrix M with linearly independent rows.
In the case of normally distributed delays the objective function of $SSPD$ has a deterministic equivalent formulation and can thus be evaluated exactly (see [2]).

3 Problem Solving Method

The idea to solve $SSPD$ (2) is the following: To search the graph for the optimal path we apply a branch-and-bound algorithm on the search space $\mathcal{P}$ of directed paths from s to t (see subsection 3.2). In order to sort out some of the subsets of $\mathcal{P}$ that do not contain the optimum we calculate lower bounds by solving the corresponding relaxed, i.e. continuous version of problem (2) (see subsection 3.1).

3.1 Solving the relaxed SSPD

First of all we remark that the continuous relaxation of problem (2) is a convex problem (see for example [2]). This allows us to solve it using a Projected Stochastic Gradient method. The basic idea is the following: at each iteration $k \geq 1$ we first calculate an estimation of the gradient $\nabla_x j(x^{k-1}, \overline{\delta}^k)$ (where x^{k-1} is the feasible solution vector computed in the previous iteration and $\overline{\delta}^k$ is a realization of the random vector δ that is regenerated at every iteration; see subsection 3.1.1). This gradient is projected on the null space of the matrix M. x^k is then computed as usual, i.e. as the sum of x^{k-1} and the projected gradient times the step size. In case we obtain negative components

of x, we adapt (i.e. shorten) the step size (subsection 3.1.2). An active set is introduced in order to handle active non-negativity constraints (subsection 3.1.3).
The final algorithm is given in Algorithm 1. In the following subsections we define the used variables and give further details on the functioning of the algorithm.

3.1.1 Estimating the gradient of j

j is differentiable everywhere except for those points x where $\sum_{a\in A}\delta(a)x_a - D = 0$. It is easy to see that the set of all these points is a null set. It can thus be neglected as the aim of all stochastic gradient algorithms is to approximate the gradient of the expectation of j via a sample procedure. Therefore, we define the gradient of j as follows:

$$\nabla_x j(x,\delta) = \begin{cases} c & \text{if } \left[\sum_{a\in A}\delta(a)x_a - D \leq 0\right] \\ c + d\cdot\delta & \text{otherwise} \end{cases}$$

3.1.2 Projection and update of x

At iteration $k \geq 1$, let $r^k := \nabla_x j(x^{k-1}, \overline{\delta}^k)$. The projection of this gradient on the null space of M is done by multiplying it with the projection matrix $T^M := \mathbb{I}_n - M^T(MM^T)^{-1}M$. Then, x is updated as follows:

$$x^k = x^{k-1} - \rho^k(T^M \cdot r^k)$$

where ρ^k is the step size given by a σ-sequence $(\rho^k)_{k\in\mathbb{N}}$ [3].
However, the predefined step size ρ^k might be too large in the sense that we can obtain components of x^k that do not lie in the unit interval.
In order to handle negative components we proceed as follows: Let I^k_- be the index set of the strictly negative components of x^k. We then compute the maximum step size that keeps x^k in the feasible region by

$$\overline{\rho}^k = \min_{i\in I^k_-}\left\{\frac{x^{k-1}_i}{(T^M\cdot r^k)_i}\right\}$$

and update x accordingly:

$$x^k = x^{k-1} - \overline{\rho}^k(T^M \cdot r^k)$$

[3] A σ-sequence is a sequence $(\rho^k)_{k\in\mathbb{N}}$ that satisfies $\lim_{k\to\infty}\rho^k = 0$ and $\sum_{k=0}^{\infty}\rho^k \to \infty$.

It follows:

Proposition 3.1 *Let x^0 be a feasible solution of the relaxed SSPD. Then, using the update procedure abovementionned, x^k remains feasible for the relaxed SSPD for all $k \geq 1$.*

3.1.3 Introducing the active set

Imagine the following situation: Let $x_i^{k-1} = 0$ for a $k \geq 1$ and an index $i \in \{1, \ldots, n\}$. After computing the projected gradient and updating x we obtain $x_i^k < 0$. Consequently, $\overline{\rho}^k = 0$, i.e. we are (and will keep) stuck on the current, probably non-optimal solution.

To prevent this, we introduce a set of additional equality constraints, the so called *active set* $\mathcal{A}^k$. As this set is continuously updated, we use a superscript that indicates in which iteration the set $\mathcal{A}^k$ is active.

For $k \geq 1$ let $I_0^{k-1} := \{i \,|\, x_i^{k-1} = 0\}$. The active set for iteration k is then defined as

$$\mathcal{A}^k = \{\, [x_i = 0] \;|i \in I_0^{k-1}\}$$

Now, instead of projecting r^k on the null space of the matrix M, we project it on the null space of a matrix Z^k: This matrix consists of the matrix M enlarged by $|I_0^{k-1}|$ rows that correspond to the equality constraints in $\mathcal{A}^k$:

$$\begin{aligned} Z_i^k &= M_i && \text{for} \quad i = 1, \ldots, n \\ Z_i^k &= e_{\tau^{k-1}(n-i)} && \text{for} \quad i = n+1, \ldots, n + |I_0^{k-1}| \end{aligned}$$

where $\{\tau^{k-1}(1), \ldots, \tau^{k-1}(|I_0^{k-1}|)\} = I_0^{k-1}$ and e_i is the $i-th$ row of the n-dimensional identity matrix. Remark that Z^k might have linearly dependent rows. In this case the projection matrix can be computed as $T^k = I - (Z^k)^+ Z^k$ where $(Z^k)^+$ is the unique Moore-Penrose pseudoinverse of Z^k.

If the computed projected gradient is zero, we have obtained an optimum of the deterministic variant of problem (2) with delay vector $\overline{\delta}^k$ and additional equality constraints given by $\mathcal{A}^k$. In this case we compute Lagrange multipliers associated with the equality constraints by solving the linear eqaution

$$Z^k (Z^k)^T \lambda = -Z^k r^k$$

If all those multipliers associated with the constraints in $\mathcal{A}^k$ are positive, we have reached the optimal solution of the deterministic variant of problem (2) with delay vector $\overline{\delta}^k$. In this case we stop the algorithm. Otherwise, we remove the constraint with the most negative multiplier from $\mathcal{A}^k$ and start a new iteration.

Algorithm 1 : Projected Stochastic Gradient Algorithm (see subsections 3.1.1-3.1.3 for (variable) definitions and further details)

- Let $(\rho^k)_{k\in\mathbb{N}}$ be a σ-sequence. Choose x^0 feasible for the $SSPD$ (2) (for example using depth-first search on the graph). Set $k = 1$.
- At iteration $k \geq 1$, define the active set $\mathcal{A}^k$. Then, for all $i \in I_0^{k-1}$, let β_i be the outcome of a Bernoulli trial with success probability p_i^k. If $\beta_i = 1$, remove the equation $[x_i = 0]$ from $\mathcal{A}^k$.
- For all $a \in A$ draw a sample $\overline{\delta}^k(a)$ of $\delta(a)$ according to its normal distribution.
- Determine Z^k and let T^k be the matrix for projection on the null space of Z^k. Compute the gradient $r^k := \nabla_x j(x^{k-1}, \overline{\delta}^k)$.
  If $\mathbf{T^k \cdot r^k = 0}$: Compute the Lagrange multipliers of the current equality constraints.
  - If all multipliers associated with the constraints in $\mathcal{A}^k$ are positive, STOP.
  - Else delete the constraint from $\mathcal{A}^k$ having the most negative associated multiplier. Set $k = k + 1$ and start a new iteration.

  Else: Update x^k as follows: $x^k = x^{k-1} - \rho^k(T^k \cdot r^k)$
  - **If** $\min_{\mathbf{i}\in\{\mathbf{1},\ldots,\mathbf{n}\}} \mathbf{x_i^k} < \mathbf{0}$: Define $I_-^k = \{i \,|\, x_i^k < 0\}$ and compute a new step size: $\overline{\rho}^k = \min_{i\in I_-^k} \left\{ \frac{x_i^{k-1}}{(T^k \cdot r^k)_i} \right\}$ Update x^k as follows: $x^k = x^{k-1} - \overline{\rho}^k(T^k \cdot r^k)$
  - Set $k = k + 1$ and start a new iteration.

However, due to the stochastic nature of our algorithm, keeping an equality constraint in $\mathcal{A}^k$ until a new optimum is found might be too restrictive: As the computation of the gradient of our objective function is based on one single realization of δ, we might "erroneously" set a component x_i^k to zero. Therefore, we further introduce for all $k \geq 1$ and for all $i \in I_0^{k-1}$ the probability p_i^k that constraint $[x_i = 0]$ is removed from $\mathcal{A}^k$ at the beginning of iteration k. Several policies for the choice of p_i^k might be considered.

3.2 Branch-and-bound framework

Definition 3.2 Let P be a directed path. We say that an arc $a = (v, w)$ *has its origin in* P, if $v \in P$ but $a \notin P$. For a path P we define the set of all arcs that have their origin on P as O_P.

The branch-and-bound algorithm can be stated as follows: First we solve the relaxed version of the overall problem, which gives us a solution of the

relaxation $\tilde{x}$ as well as a first lower bound LB. We then begin to search for a feasible binary solution by plunging the graph (see phase 4). The obtained directed path P from s to t together with the corresponding lower bound $LB(P) = LB$ are stored in the list of waiting s-t-paths $\mathcal{L}$. In addition we store the value of $\tilde{x}$ for all arcs $a \in O_P$ in a variable $x_P(a)$. The solution value of $SSPD$ given by P is our first upper bound and it is stored in the variable UB. Then, each further iteration of our branch-and-bound algorithm consists of (up to) five phases:

Phase 1: Selecting a branch
If $\mathcal{L}$ is empty, the algorithm terminates. The current value of UB is the optimal solution value.
Otherwise, we select a path $P \in \mathcal{L}$ such that $LB(P) = \min_{Q \in \mathcal{L}} LB(Q)$.

Phase 2: Selecting an arc
If no arc in O_P is left that has not already be examined (i.e. $\max\{x_P(b)|b \in O_P\} = -1$, see Phase 5), we delete P from $\mathcal{L}$, end the iteration and go back to phase 1. Otherwise, we propose two different policies:

a) We choose the arc $a \in O_P$ with $x_a = \max\{x_P(b)|b \in O_P\}$.

b) We go to the first vertex v on P such that there still exists at least one arc $(v, w) \in O_P$ that has not already be examined (i.e. such that $\max\{x_P(b)|\exists w \in A : b = (v, w)\} \neq -1$). We then choose the arc a such that $x_P(a) = \max\{x_P(b)|\exists w \in A : b = (v, w)\}$.

If adding a to the subpath s-P-v leads to a non-feasible solution, we reject a and choose another arc in O_P following either policy a) or policy b).

Phase 3: Calculating a lower bound
Let $a = (v, w)$ be the arc chosen in phase 2. Consider the relaxed subproblem of $SSPD$ obtained by fixing the first part of the s-t-path to s-P-(v, w). Solving this subproblem gives us a lower bound $\widetilde{LB}$ on its binary solution and a solution vector $\tilde{x}$. If $\widetilde{LB} < UB$ we go to phase 4. Otherwise, we reject a and choose a new arc (phase 2).

Phase 4: Plunging
In order to find a new s-t-path P' containing the sub-path s-P-(v, w) different policies might be considered, for example:

i) Starting from vertex w, we always add the outgoing arc with the highest value of $\tilde{x}$.

ii) We continuously update the objective function value of the sub-path $\tilde{P}$ already chosen. Arrived at a new vertex u, we compute for all $u' \in V$ with $(u, u') \in A$ the objective function value for the sub-path $s - \tilde{P} - (u, u')$ and choose the subpath with the highest value.

One could also consider a randomized choice of the next arc to add: As the sum of the $\tilde{x}$-values of the outgoing arcs is always one, we could interpret these values to be the probability that the arc is chosen next.

Phase 5: Storage
Path P' is stored together with the corresponding lower bound $LB(P') = \widetilde{LB}$ in the list of waiting paths $\mathcal{L}$. In addition, we define for all arcs $a \in O_{P'}$ the value $x_{P'}(a)$ as follows: Previously examined or rejected arcs with origin in P' are assigned the value -1. For those non-examined arcs a that have their origin on s-P-v $x_{P'}(a)$ is set to $x_P(a)$. For the rest of the outgoing arcs, we store the corresponding component of $\tilde{x}$.
If the solution value of the $SSPD$ given by P' is higher than the current upper bound UB, we update UB.

4 Conclusion

In this paper we solve a stochastic version of the restricted shortest path problem. We propose to use a branch-and-bound framework to search the set of feasible paths. Lower bounds are obtained by solving the corresponding convex relaxations which in turn is done using a Projected Stochastic Gradient algorithm involving an active set method.

References

[1] H. C. Joksch. The shortest path problem with constraints. *Journal of Mathematical Analysis and Applications*, 14:191–197, 1966.

[2] Stefanie Kosuch and Abdel Lisser. Upper bounds for the 0-1 stochastic knapsack problem and a b&b algorithm. *Annals of Operations Research (Online First)*, 2009. http://dx.doi.org/10.1007/s10479-009-0577-5.

[3] Bram Verweij, Shabbir Ahmed, Anton J. Kleywegt, George Nemhauser, and Alexander Shapiro. The sample average approximation method applied tostochastic routing problems: A computational study. *Computational Optimization and Applications*, 24(2-3):289 – 333, 2003.