Maximum Probability Shortest Path Problem

Jeremy Krebs - Guillaume Soulié

Université Paris Saclay

6 novembre 2017

Shortest Path is a known problem and has many applications in "real life".

- Goods transport (industrial and private)
- Food Delivery (Deliveroo - Foodora - UberEats)

As this problem is well known, lots of scientists presented their work on related subject, most of the times with different hypotheses :

As this problem is well known, lots of scientists presented their work on related subject, most of the times with different hypotheses :

- Without resource constraints

As this problem is well known, lots of scientists presented their work on related subject, most of the times with different hypotheses :

- Without resource constraints
- With deterministic resource constraints

As this problem is well known, lots of scientists presented their work on related subject, most of the times with different hypotheses :

- Without resource constraints
- With deterministic resource constraints
- With stochastic resource constraints

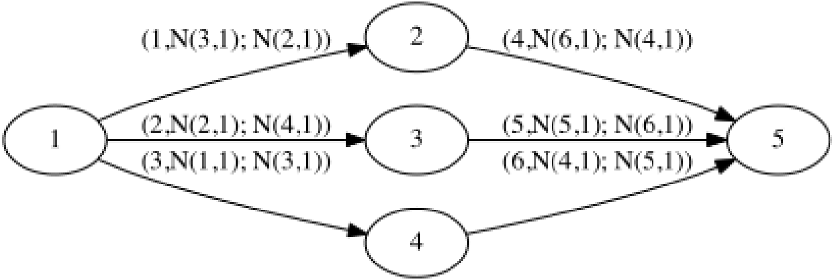
As this problem is well known, lots of scientists presented their work on related subject, most of the times with different hypotheses :

- Without resource constraints
- With deterministic resource constraints
- With stochastic resource constraints

or also with a different optimization problem like utility functions to maximize or cost functions to minimize.

- Graph with weights on arcs, source node s and sink node t,
- Stochastic resource consumptions with normal distribution,
- K resources,
- Threshold C of the cost function (maximum allowed weight of the path).

- Graph with weights on arcs, source node s and sink node t,
- Stochastic resource consumptions with normal distribution,
- K resources,
- Threshold C of the cost function (maximum allowed weight of the path).



SRCSP can be formulated as this optimization problem :

$$max\ \mathbf{Pr}\{\tilde{a}_k^T x \leq d_k, k = 1..K\}$$

$$\begin{aligned} s.t.\ c^T x &\leq C \\ Mx &= b \\ x &\in \{0,1\}^n \end{aligned}$$

where :

- $x(e) = 1$ if $x(e) \in$ path P
- The a_k are multi-variate vectors with mean μ_k and known covariance matrix V_k
- M is the node-arc incidence matrix. $M(i,e) \in \{-1, 0, 1\}$
- b is a vector with 0 everywhere except $b(s) = 1$ and $b(t) = -1$

SRCSP can be reformulated as :

$$
\begin{aligned}
&max\ \ p\\
&s.t.\ \ p \leq \mathbf{Pr}\{\tilde{a}_k^T x \leq d_k, k = 1..K\}\\
&\qquad c^T \leq C\\
&\qquad Mx = b\\
&\qquad x \in \{0,1\}^n
\end{aligned}
$$

Branch and bound framework

Branch and bound framework

- find the upper bound

Branch and bound framework

- find the upper bound
- find the optimum, using branch and bound

Branch and bound framework

- find the upper bound - **relaxation problem**
- find the optimum, using branch and bound

With $K = 1$ we have :

$$
\begin{aligned}
& max\ p \\
& s.t.\ p \leq \mathbf{Pr}\{\tilde{a}_1^T x \leq d_1\} \\
& \qquad c^T \leq C \\
& \qquad Mx = b \\
& \qquad x \in \{0,1\}^n
\end{aligned}
$$

Using the known multivariate distribution parameters we get :

$$\begin{aligned} & max\ p \\ s.t.\ & F^{-1}(p)(x^T V_1 x)^{\frac{1}{2}} \leq d_1 - \mu_1^T x \\ & c^T \leq C \\ & Mx = b \\ & x \in \{0, 1\} \end{aligned}$$

Relaxing the problem we get (SRCSPI) with $p \leq \frac{1}{2}$:

$$\begin{aligned} &max\ 0 \\ &s.t.\ F^{-1}(p)(x^T V_1 x)^{\frac{1}{2}} \leq d_1 - \mu_1^T x \\ &\quad c^T \leq C \\ &\quad Mx = b \\ &\quad 0 \leq x \leq 1 \end{aligned}$$

We can solve it using the binary search procedure. We take $p_1 \leq \frac{1}{2}$ a feasible solution, a lower bound of SRCSP. p_l and p_u are lower and upper bounds of SRCSPI. Then we iterate :

We can solve it using the binary search procedure. We take $p_1 \leq \frac{1}{2}$ a feasible solution, a lower bound of SRCSP. p_l and p_u are lower and upper bounds of SRCSPI. Then we iterate :

Start $p_l = \frac{1}{2}$ and $p_u = 1$. Iteration counter $t = 1$

We can solve it using the binary search procedure. We take $p_1 \leq \frac{1}{2}$ a feasible solution, a lower bound of SRCSP. p_l and p_u are lower and upper bounds of SRCSPI. Then we iterate :

Start $p_l = \frac{1}{2}$ and $p_u = 1$. Iteration counter $t = 1$

Search Solve SRCSPI with $p = p_t$. If SRCSPI has an optimal solution, set $p_l = p_t$, otherwise $p_u = p_t$.

We can solve it using the binary search procedure. We take $p_1 \leq \frac{1}{2}$ a feasible solution, a lower bound of SRCSP. p_l and p_u are lower and upper bounds of SRCSPI. Then we iterate :

Start $p_l = \frac{1}{2}$ and $p_u = 1$. Iteration counter $t = 1$

Search Solve SRCSPI with $p = p_t$. If SRCSPI has an optimal solution, set $p_l = p_t$, otherwise $p_u = p_t$.

Stop Stop when $\frac{p_u - p_l}{2} \leq \epsilon$. Otherwise $t++$ and $p_t = \frac{p_l + p_u}{2}$

We now consider $K > 1$.
We can formulate the joint probabilistic SRCSP as follow :

$$
\begin{aligned}
& max \;\; p \\
& s.t. \;\; p \leq \mathbf{Pr}\{\tilde{a}_k^T x \leq d_k, \;\; k = 1, .., K\} \\
SRCSPJ \;\; & c^T \leq C \\
& Mx = b \\
& 0 \leq x_i \leq 1, \;\; i = 1, ..., n
\end{aligned}
$$

We now consider $K > 1$.
We can formulate the joint probabilistic SRCSP as follow :

$$\begin{aligned} & max \ p \\ & s.t. \ p \leq \mathbf{Pr}\{\tilde{a}_k^T x \leq d_k, \ k = 1, .., K\} \\ SRCSPJ \ & c^T \leq C \\ & Mx = b \\ & 0 \leq x_i \leq 1, \ i = 1, ..., n \end{aligned}$$

If p is greater than a threshold value (theoreme 4.0.1), M(p) is convex.

Deterministic reformulation of SRCSPJ :

$$
\begin{aligned}
& max\ \ p \\
& s.t.\ F^{-1}(p^{y_k})(x^T V_k x)^{\frac{1}{2}} \leq d_k - \mu_k^T x,\ \ k = 1, .., K \\
& \quad \sum_{k=1}^{K} y_k = 1,\ \ y_k \geq 0,\ \ k = 1, ..., K \\
& \quad c^T x \leq C \\
& \quad Mx = b,\ \ x \in \{0, 1\}^n.
\end{aligned}
$$

Main idea

Main idea

- Firstly we approximate $F^{-1}(p_0^k)$ with a piecewise tangent approximation of y_k.
  Afterwards, we get an approximation, which is SOCP problem.

Main idea

- Firstly we approximate $F^{-1}(p_0^k)$ with a piecewise tangent approximation of y_k.
  Afterwards, we get an approximation, which is SOCP problem.
- Secondly, we solve the SOCP problem, whose optimal value is an upper bound of SRCSP.

Main idea

- Firstly we approximate $F^{-1}(p_0^k)$ with a piecewise tangent approximation of y_k.
  Afterwards, we get an approximation, which is SOCP problem.
- Secondly, we solve the SOCP problem, whose optimal value is an upper bound of SRCSP.
- Finally, we found the optimum with branch and bound algorithm.

Finally, we can approximate it with the following convex relaxation :

$$
\begin{aligned}
\min \sum_{k=1}^{K} y_k & \\
s.t.\ & (\tilde{z}_k^T V_k \tilde{z}_k)^{\frac{1}{2}} \leq d_k - \mu_k^T x,\ \ k = 1, ..., K \\
& \tilde{z}_{ki} \geq a_j x_i + b_j y_{ki},\ \ j = 0, ..., n,\ \ i = 1, ..., n \\
& 0 \leq y_k \leq -\log_{p_0}(2),\ \ k = 1, ..., K \\
& 0 \leq y_{ki} \leq y_k,\ \ y_{ki} \geq y_k + x_i - 1,\ \ i = 1, ..., n,\ \ k = 1, ..., K \\
& c^T x \leq C,\ \ Mx = B,\ 0 \leq x_i \leq 1,\ \ i = 1, ..., n \\
& M\tilde{y}_k = y_k b,\ \ k = 1, ..., K
\end{aligned}
$$

where $\tilde{y}_k = (y_{k1}, ..., y_{kn})$

We formulate this problem taking individual probabilities constraints :

$$
\begin{aligned}
& max\ \ p \\
s.t.\ \ & p \leq \mathbf{Pr}\{\tilde{a}_k^T x \leq d_k\},\ \mathbf{k=1,..,K} \\
& c^T \leq C \\
& Mx = b \\
& x \in \{0,1\}^n
\end{aligned}
$$

We can relax the equation as we did before to get (RSRCSPJI) :

$$
\begin{aligned}
& max\ 0 \\
& s.t.\ F^{-1}(p)(x^T V_k x)^{\frac{1}{2}} \leq d_k - \mu_k^T x,\ \ k = 1, .., K \\
& \quad c^T \leq C \\
& \quad Mx = b \\
& \quad 0 \leq x_i \leq 1,\ \ i = 1, .., n
\end{aligned}
$$

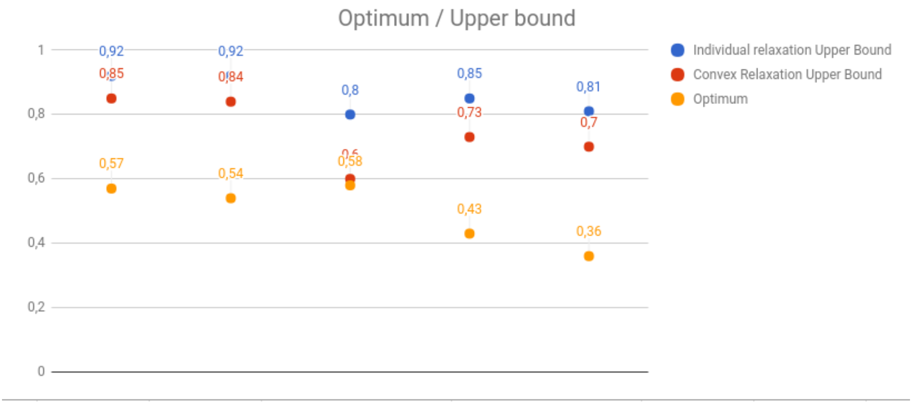
We use the relaxed equation using the Binary Search Procedure again :

Start $p_l = \frac{1}{2}$ and $p_u = 1$. Iteration counter $t = 1$

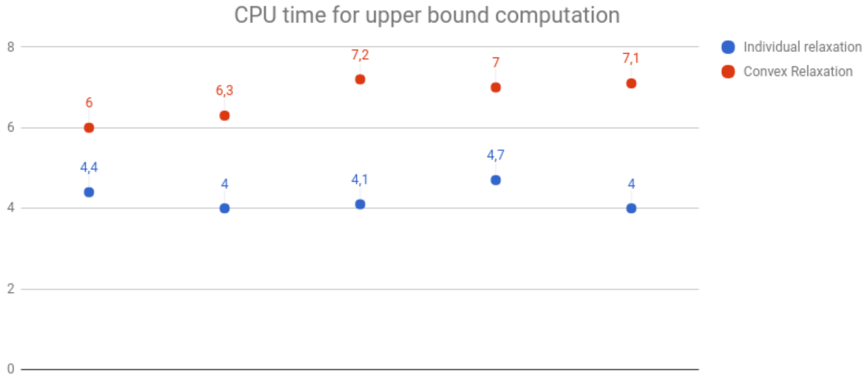
Search Solve RSRCSPJI with $p = p_t$. If RSRCSPJI has an optimal solution, set $p_l = p_t$, otherwise $p_u = p_t$.

Stop Stop when $\frac{p_u - p_l}{2} \leq \epsilon$. Otherwise $t++$ and $p_t = \frac{p_l + p_u}{2}$

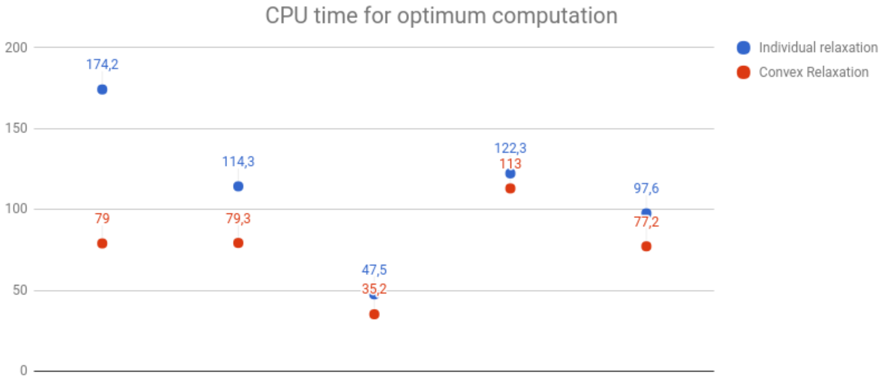
- upper bound is more accurate



- Individual relaxation upper bound computation is faster



- convex relaxation optimum computation is faster (upper bound is smaller)



Graphs kind

- $(n, m) = (23, 40)$ **(chart)**.

Graphs kind

- $(n, m) = (23, 40)$ **(chart)**.
- $(n, m) = (50, 413)$.

Graphs kind

- $(n, m) = (23, 40)$ **(chart)**.
- $(n, m) = (50, 413)$.
- random graphs with $n = 50$.

Graphs kind

- $(n, m) = (23, 40)$ **(chart)**.
- $(n, m) = (50, 413)$.
- random graphs with $n = 50$.

Method easily extendable to solve larger size instances.

Conclusion

Conclusion

- the paper proposes a branch and bound framework to solve the stochastic resource constrained shortest path

Conclusion

- the paper proposes a branch and bound framework to solve the stochastic resource constrained shortest path
- it introduces a convex relaxation to find the upper bound.

Conclusion

- the paper proposes a branch and bound framework to solve the stochastic resource constrained shortest path
- it introduces a convex relaxation to find the upper bound.
- it shows by experiments that this convex relaxation outperforms indivudal relaxation.

Conclusion

- the paper proposes a branch and bound framework to solve the stochastic resource constrained shortest path
- it introduces a convex relaxation to find the upper bound.
- it shows by experiments that this convex relaxation outperforms indivudal relaxation.

Going further

- generalize to dependent random variable

Conclusion

- the paper proposes a branch and bound framework to solve the stochastic resource constrained shortest path
- it introduces a convex relaxation to find the upper bound.
- it shows by experiments that this convex relaxation outperforms indivudal relaxation.

Going further

- generalize to dependent random variable
- study the influence of some variable (for exemple N, the number of segment in tangent approximation, on the CPU time / upper bound value).

Conclusion

- the paper proposes a branch and bound framework to solve the stochastic resource constrained shortest path
- it introduces a convex relaxation to find the upper bound.
- it shows by experiments that this convex relaxation outperforms indivudal relaxation.

Going further

- generalize to dependent random variable
- study the influence of some variable (for exemple N, the number of segment in tangent approximation, on the CPU time / upper bound value).
- find a lower bound to give the branch and bound algorithm with another approximation of $F^{-1}(p_0^k)$.

Questions ?