Optimal trip planning in transport systems with random delays

Keith Briggs & Peter Kin Po Tam

Mobility Research Centre, BT Innovate & Design
http://keithbriggs.info

Open University, Mathematics Dept
2010-05-25 1400

BT Research - Adastral Park

Optimizing departure time

The primary problem:

- Given:
  1. The time a passenger would like to reach the destination
  2. The probability that a passenger would like to reach the destination on time (a high probability if it is a time-critical trip or a smaller probability if it is less important)
- Find:
  1. The best route
  2. The latest time the passenger should start from the first train station

Let

- τ be the time the passenger would like to reach the destination
- ϵ be the probability that the passenger would like to reach the destination "on time"
- t_d be the time the passenger departs from the first station
- T_a be the time the passenger arrives at the destination (arrival time)

Then the optimization problem is:

$$\begin{array}{rl} \text{max} & t_d \\ \text{subject to} & \text{Prob}[T_a > \tau] < 1 - \epsilon \end{array}$$

Modelling train delay

In order to calculate Prob$[T_a > \tau]$, we will need to solve the following secondary problem:

- Given:
  1. A route
  2. A timetabled service for each train
  3. A model of the distribution of delays for each train
  4. A model of the distribution of starting time of the passenger
- Find:
  1. The probability distribution of the arrival time T_a for the passenger at the destination

Assumptions and notations

Assumptions:

- The departure times of any two trains are statistically independent
- The order of the departure for trains may vary due to delay
- When changing trains, passengers always catch the first train that departs to their next station on their chosen route

Notation:

$X_i \sim f_i$ probability distribution function (pdf)

F_i cumulative distribution function (cdf)

$$F_i(t) = \int_{-\infty}^{t} f_i(x)dx$$

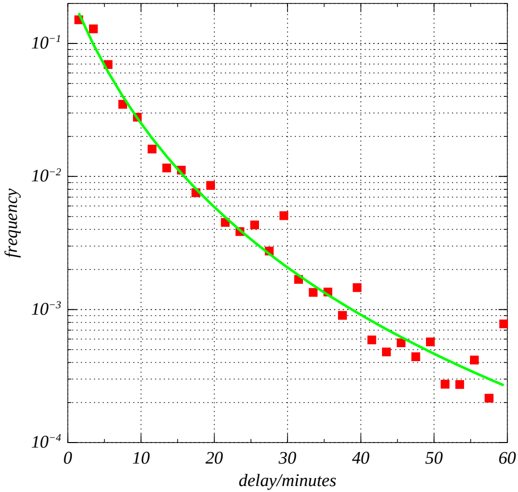
$$[\![x \geqslant t]\!] = \begin{cases} 1 & \text{if } x \geqslant t, \\ 0 & \text{otherwise} \end{cases}$$

The q-exponential law

- Exponential law: $f_\beta(x) \propto \exp(-\beta x)$
- $e_{r,\beta}(x) := Z(1+\frac{\beta x}{r})^{-r}$, $\beta > 0$, $r > 1$
- $Z := \beta(1-\frac{1}{r})$
- mean $\mu := \frac{1}{\beta(1-\frac{2}{r})}$
- $\lim_{r\to\infty} e_{r,\beta}(x) = Z\exp(-\beta x)$
- small r gives a power-law (long tail)
- The departure times for every trains can be modelled by q-exponential distribution by some parameters β and r.

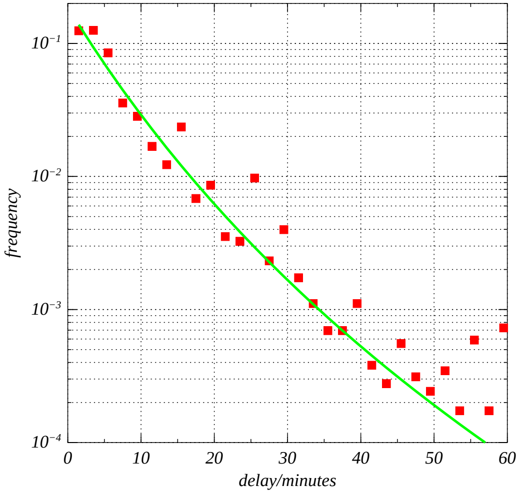
Birmingham, all departures

$r = 3.8$
$\beta = 0.33$



Coventry to Birmingham

$r = 9.0$
$\beta = 0.21$



The discrete $q-$exponential model

- The discrete $q-$exponential model is more suited for the train model since the departure times of trains are considered in minutes
- Bins have length $\mathrm{d}t$, typically one minute
- The values for each bin are calculated using the CDF of the continuous model, i.e. $\int_a^{a+\mathrm{d}t} e_{r,\beta}(x)\ \mathrm{d}x$ where a is the value of the bin
- The distribution is truncated; the departure delay cannot be <0 or greater than some maximum

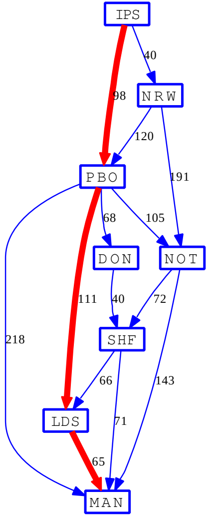
Stochastically short paths

- Given a graph with RVs as edge weights and two nodes, we could:
  - minimize expected time to travel between the nodes
  - find a route which maximizes the probability that it is shortest
  - find the route of shortest mean time, subject to some condition on the variance
  - ...

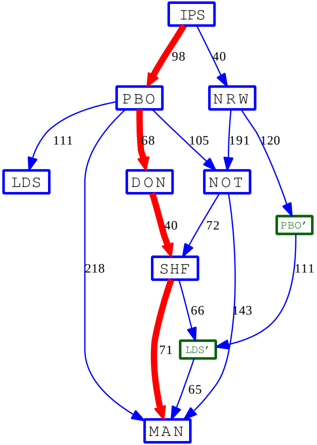
The problem formalized

- Given: a weighted digraph g, a timetable $\mathsf{TT}(n_0, n_1)$ for each arc $(n_0, n_1) \in g$, a final arrival time T_a, and parameters $\tau > 0, \epsilon > 0$.
- To find: a route ρ and maximal departure time t such that Prob[arrival after $\alpha + \tau$] $< \epsilon$

Short paths in a weighted digraph

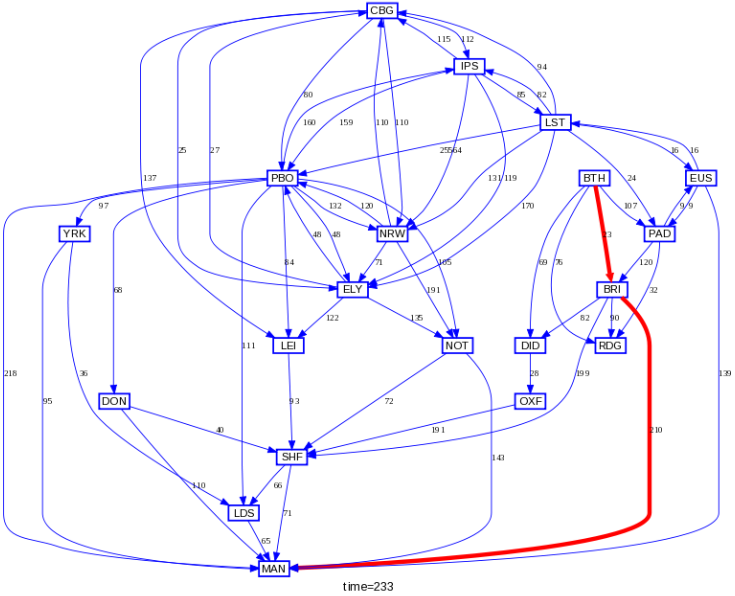


time=275

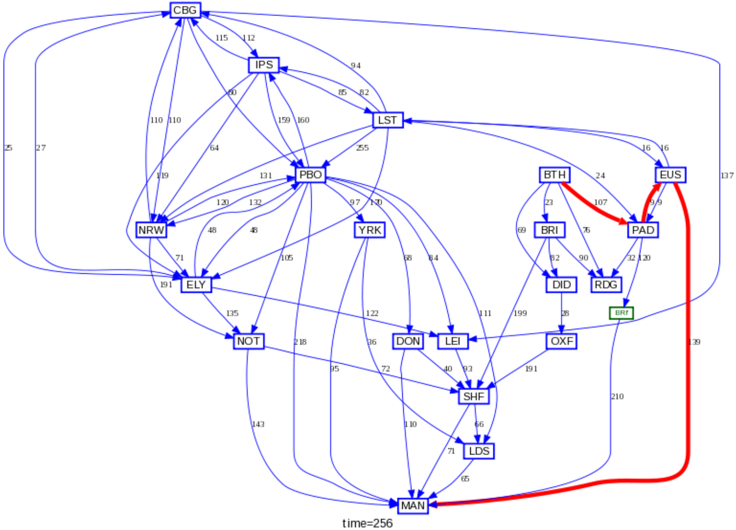


time=278

Bath to Manchester, shortest mean time



Bath to Manchester, second shortest mean time



Transformation using kernel

- The departure time of the passenger at the initial station is modelled as a probability distribution
- We need to compute the probability distribution model for the arrival time of the passenger at the next station

$$A_0 \xrightarrow[K_0]{} A_1 \xrightarrow[K_1]{} \ldots \xrightarrow[K_{r-1}]{} A_r$$

A_i is the probability distribution of arrival time i.

K_i represents the set of probability distributions of departure times of each train at station i and we will call it the *kernel*.

Aside: order statistics

Let $X_i \sim f$, $i = 0, \ldots, n-1$ be iid. Define $X_{(0)}$ to be the minimum X_i. We compute the pdf $f_{(0)}$ of $X_{(0)}$:

$$
\begin{aligned}
f_{(0)}(x)\mathrm{d}x &= \mathrm{Prob}[X_{(0)} \in \mathrm{d}x] \\
&= \mathrm{Prob}[\text{one } X_i \in \mathrm{d}x,\ \text{others} > x] \\
&= \sum_i \mathrm{Prob}[X_i \in \mathrm{d}x, \underset{j \neq i}{X_j} > x] \\
&= \sum_i \mathrm{Prob}[X_i \in \mathrm{d}x]\,\mathrm{Prob}[\underset{j \neq i}{X_j} > x] \\
f_{(0)}(x) &= \sum_i f(x) \prod_{j \neq i} (1 - F(x)) \\
&= n f(x)(1 - F(x))^{n-1}
\end{aligned}
$$

The continuous time model

Let $K_{(0|t)}(x)$ be the probability density of departing at time x given that the passenger arrives at time t.

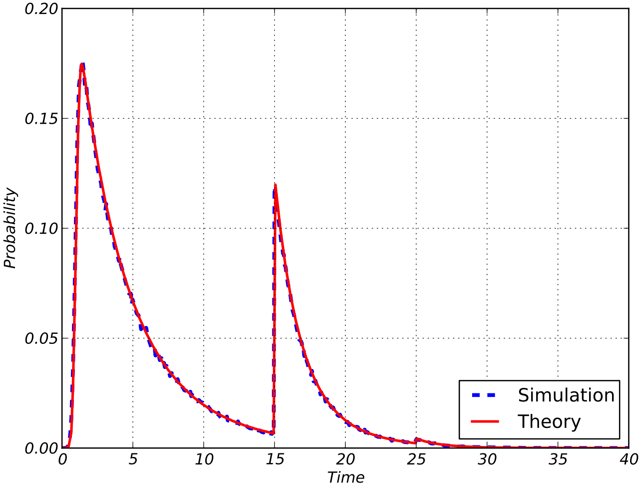
$$\begin{aligned}
K_{(0|t)}(x)\mathrm{d}x &= [\![x>t]\!]\,\mathrm{Prob}[X_{(0|t)} \in \mathrm{d}x] \\
&= [\![x>t]\!]\,\mathrm{Prob}[\text{one } X_i \in \mathrm{d}x,\ \text{others } > x \text{ or } \leqslant t] \\
&= [\![x>t]\!] \sum_i \mathrm{Prob}[X_i \in \mathrm{d}x, \underset{j\neq i}{X_j} > x \text{ or } \leqslant t] \\
&= [\![x>t]\!] \sum_i \mathrm{Prob}[X_i \in \mathrm{d}x]\,\mathrm{Prob}[\underset{j\neq i}{X_j} > x \text{ or } \leqslant t] \\
K_{(0|t)}(x) &= [\![x>t]\!] \sum_i f_i(x) \prod_{j\neq i} (1 - F_j(x) + F_j(t))
\end{aligned}$$

Integral transforms

$$A_n(x) = \int_{-\infty}^{\infty} A_{n-1}(t) K_{(n-1|t)}(x) \mathrm{d}t, \ n = 1, 2, \ldots$$

- Exact results (and agree with simulation)
- Integrals cannot be done analytically
- The amount of computer time needed to calculate $A_i(x)$ increases dramatically as i increases

Continuous model simulation for catching one train



The discrete time model

Let T be the set of all trains, t be the time the passenger arrives at the station and x be the time the passenger departs.

$$\begin{aligned} K_{(0|t)}(x) &= [\![x > t]\!] \text{Prob}[X_{(0|t)} = x] \\ &= [\![x > t]\!] \text{Prob}[\text{at least one } X_i = x,\ \text{others } > x \text{ or } \leqslant t] \end{aligned}$$

Let T be the set of trains departing after time t; then

$$\begin{aligned} K_{(0|t)}(x) &= [\![x > t]\!] \sum_{\emptyset \neq S \subseteq T} \text{Prob}[\underset{i \in S}{X_i} = x, \underset{j \notin S}{X_j} > x \text{ or } \leqslant t] \\ &= [\![x > t]\!] \sum_{\emptyset \neq S \subseteq T} \text{Prob}[\underset{i \in S}{X_i} = x] \text{Prob}[\underset{j \notin S}{X_j} > x \text{ or } \leqslant t] \\ &= [\![x > t]\!] \sum_{\emptyset \neq S \subseteq T} \prod_{i \in S} f_i(x) \prod_{j \notin S} (1 - F_j(x) + F_j(t)) \end{aligned}$$

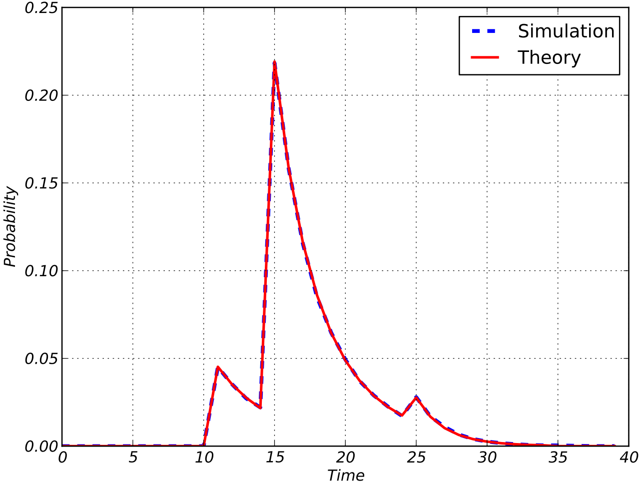
Matrix multiplication

Let $K_0 = k_{ij}^{(0)} = (K_{(0|j)}(i))$, then

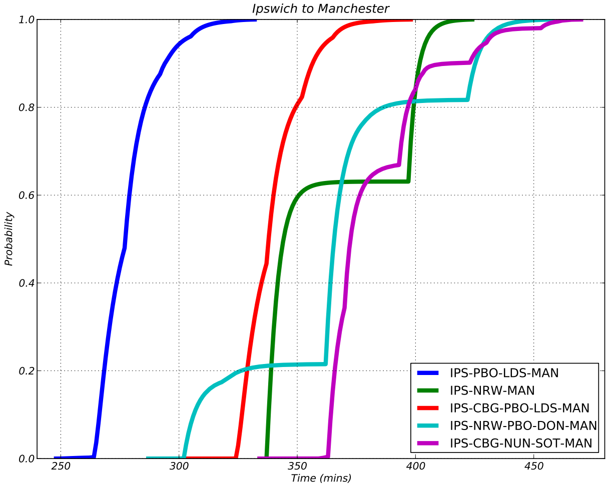
$$A_1 = K_0 A_0$$
$$A_2 = K_1 A_1 = K_1 K_0 A_0$$
$$\ldots$$
$$A_n = K_{n-1} \ldots K_0 A_0$$

- Kernel and the distribution of arrival time can be considered as matrix and vector
- Finding the distribution of the arrival time at the next station is exactly the same as matrix multiplication
- Matrix calculations are generally much quicker than integration

Discrete model simulation for catching one train



CDF for journey times from Ipswich to Manchester



Gray code method

- The Gray code method is a method of generating a list of subsets such that adjacent subsets in the list differ only by a single element
- Using the Gray code method, we can generate all the subsets with the same cardinality
- Since after each iteration the new set generated differs by a single element from the previous set, we can use the product obtained from the previous iteration to calculate the next product
- There is no need to multiply all the products again; we only need to divide (element taken out from the previous set) and multiply (element put in the new set) once and we will obtain the next product

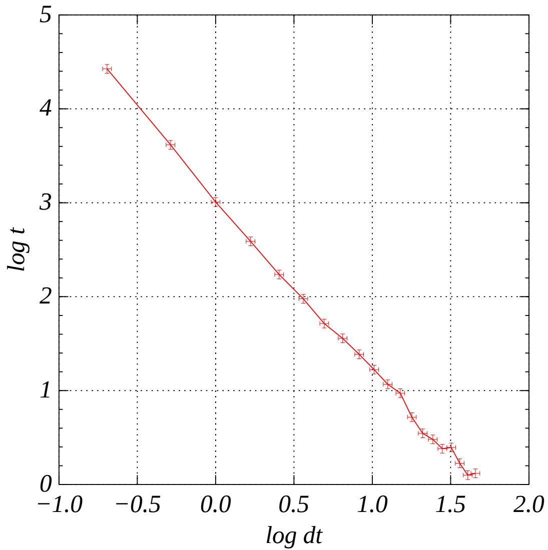
Changing the length of each interval

- By default, the probabilities are calculated for every minute, i.e. each interval is one minute.
- For a long journey, the amount of calculations will increase significantly.
- The passengers may prefer to know the probability of arriving at their destinations at some arbitrary interval.

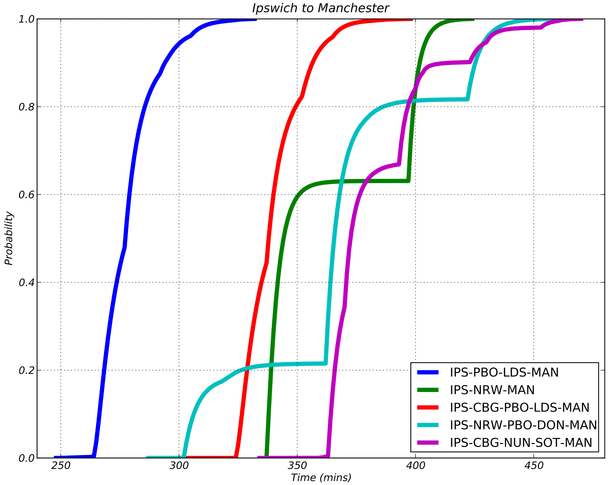
Let t be the CPU time for the calculation of each journey and dt be the length of each interval, then

- $\frac{\log t}{\log dt} \sim 2$ or $t \sim dt^2$

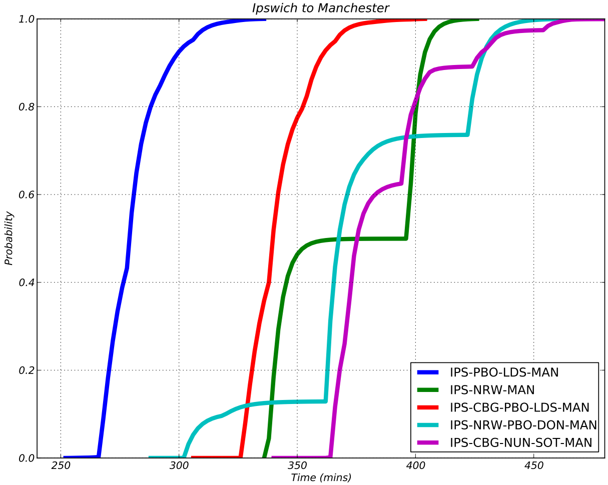
CPU time needed with different interval length



CDF for journey from Ipswich to Manchester with interval length 1 when departure at time 0



CDF for journey from Ipswich to Manchester with interval length 2 when departure at time 0



Calculating $K_{(0|t)}(x)$

- The total numbers of non-empty subsets of T is $2^m - 1$ where m is the cardinality of T
- The number of operations is exponential
- Some or maybe most of the terms in the sum are zero
- The number of calculations can be reduced if we can find these zero terms
- The number of operations can also be reduced by reducing some degree of accuracy to the probability

Finding the zero terms

Let S be any non-empty subset of T and

$$f^S(x) = \prod_{i \in S} f_i(x) \text{ and } g^S(x,t) = \prod_{j \notin S} g_j(x,t) = \prod_{j \notin S} (1 - F_j(x) + F_j(t)).$$

Then

$$K_{(0|t)}(x) = \sum_{\emptyset \neq S \subseteq T} f^S(x) g^S(x,t).$$

- The product is zero if any factor in either $f^S(x)$ or $g^S(x,t)$ is zero
- Since the order of multiplication and addition will not affect the result, we can sort $f_i(x)$ in descending order and relabel the indices

- Find all the i such that $f_i(x) \neq 0$ and let the collection of indices be the set U
- Find all the $j \in U$ such that $g_j(x,t) \neq 0$ and let the collection of indices be the set V

Then the set T can be partitioned into the following subsets, V, $U - V$ and $T - U$ and we can obtain the following table:

$i \in$	V	$U - V$	$T - U$
$f_i(x)$	>0	>0	$=0$
$g_i(x,t)$	>0	$=0$	$\geqslant 0$

From the table, we will have the following observations:

- If $\exists i \in T - U$ such that $g_i(x,t) = 0$, then either $f^S(x)$ or $g^S(x,t)$ is zero for all S and hence $K_{(0|t)}(x) = 0$
- If $f^S(x)g^S(x,t) \neq 0$, it immediately implies that $U - V \subseteq S$
- Similarly, if $f^S(x)g^S(x,t) \neq 0$, it immediately implies that $T - U \subseteq T - S$
- Although the number of operation is still exponential, the total number of subsets is reduced

Controlling the error

- Without loss of generality, we can assume that $f^S(x)g^S(x,t) \neq 0, \ \forall S$
- First consider all possible subsets of T with cardinality one and calculate the sum, i.e. $\sum_{|S|=1} f^S(x)g^S(x,t)$
- Calculate the upper bounds by consider the following:

$$\text{upper bound} = \begin{array}{l}\text{remaining number of subsets } \times \\ \text{upper bound on products } f^S(x)g^S(x,t), \ |S|>1\end{array}$$

- If the upper bound is less than some satisfactory value, i.e. ϵ (absolute error) or $\epsilon \times$sum (relative error), then the sum is accepted
- Otherwise, consider all subsets with cardinality one and two and repeat the process until the upper bound is less than ϵ

Increasing the speed of matrix multiplication

- The number of rows (i) needed is not known in general
- While calculating d_{l+1}, instead of calculating the whole matrix K_l by using random number of rows, we will calculate the first row of K_l and find the first element of d_{l+1}
- Then we will calculate the second row of K_l, find the second element of d_{l+1} and repeat the process.
- Since d_{l+1} is a probability distribution, therefore the sum of all element in d_{l+1} is equal to 1
- Using this fact, we will calculate the sum after every step and we will stop when the sum is equal or relatively close to 1
- This method not only solve the problem of not knowing the number of rows required, it also uses less memory since no large matrix is calculated and saves compution time, as only required rows are calculated.

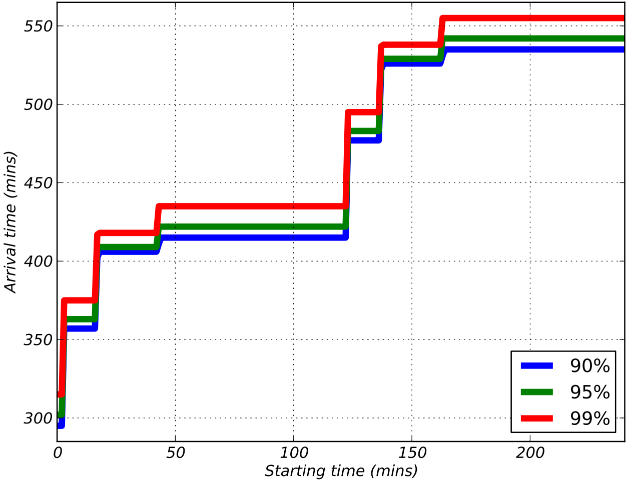
Finding the optimized departure time

- From the CDF we can calculate the time a passenger will arrive the destination at a given probability for a fixed departure time
- We would like to find the latest departure time such that the passenger will arrive the destination on time at a probability given by the passenger
- The optimized departure can be found by using bisection method

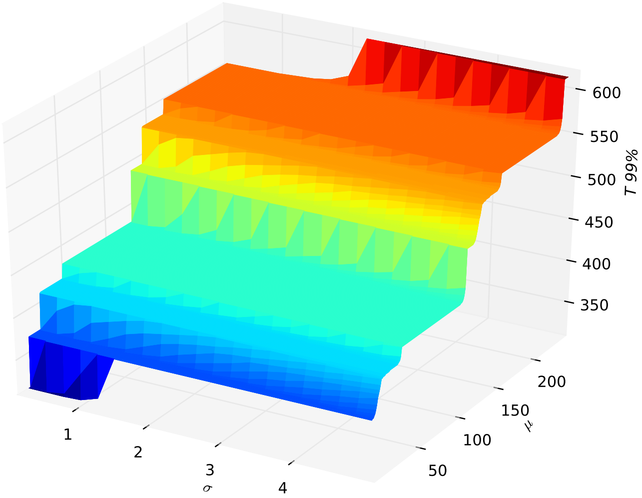
Outline of the full algorithm

1. Obtain two departure times, x, y such that $x < y$ and obtain the arrival time t_x, t_y using the CDF of departure time x and y respectively.
2. If the arrival time set by the passenger T is not between t_x and t_y, then obtain a new x and y accordingly
3. Find the midpoint z of x and y
4. Calculate the CDF for departure time z and obtain the arrival time t_z
5. If $t_x \leqslant T \leqslant t_z$, set $y := z$, otherwise set $x := z$
6. Repeat the process until $y - x$ is sufficiently small

Arrival time for journey from Ipswich to Manchester with different starting time



Arrival time for journey from Ipswich to Manchester with starting times as Gauss function



Future work

- Arrival time distribution
- Investigate the correlation between departure time and arrival time distribution

References

- K M Briggs & C Beck, *Modelling train delays with q-exponential functions* Physica **A 378**, 498–504 (2007).
- H A David, *Order statistics* Wiley (1970).
- J. Loughry & J.I. van Hemerty & L. Schoofsz *Efficiently Enumerating the Subsets of a Set* http://applied-math.org/subset.pdf, (2000).
- F. Ruskey *Combinatorial Generation* http://www.1stworks.com/ref/RuskeyCombGen.pdf, (2003)
- D. Sethi, *Optimal railway-trip planning* University of York, (2009).