

Genetic Algorithm for determination of event collision time and particle identification by Time-Of-Flight at SPD NICA

Semyon Yurchenko

Saint Petersburg State University

sem2600@mail.ru

November 23, 2022

Using information about particles trajectories and hits from TOF detector determine time of pp -collision.

- 1 In this work only tracks with momentum below 500 MeV will not be considered.
- 2 Resolution of TOF detector $\sigma_t = 70$ ps.
- 3 Momentum resolution: $\frac{\sigma_p}{p} = 2\%$
- 4 TOF radius is 1 m and length of 3.772 m.

Plan of simulation

- 1 Get tracks of charged particles with momentum over 500 MeV from Pythia8 events with $\sqrt{s} = 27 \text{ GeV}$.
- 2 Calculate intersection point with TOF detector(t_i).
- 3 Calculate arc length of trajectory(L_i).
- 4 Smear t_i with $N(t_i, 70 \text{ ps})$ and p_i with $N(p_i, 0.02 \cdot p_i)$.
- 5 Using information about arc lengths of trajectories, TOF hits and particle momentum determine time of pp -collision(t_0).

- 1 Get tracks of charged particles with momentum over 500 MeV from Pythia8 events with $\sqrt{s} = 27 \text{ GeV}$.
- 2 Calculate intersection point with TOF detector(t_i).
- 3 Calculate arc length of trajectory(L_i).
- 4 Smear t_i with $N(t_i, 70 \text{ ps})$ and p_i with $N(p_i, 0.02 \cdot p_i)$.
- 5 **Using information about arc lengths of trajectories, TOF hits and particle momentum determine time of pp -collision(t_0).**

How brute force works:

- 1 Choose particles types to make tof hypotheses $\rightarrow [\pi^\pm, K^\pm, p^\pm]$.

$$tof_{ik} = \frac{L_i}{c} \sqrt{1 + \frac{m_k^2}{p_i^2}} \quad (1)$$

- 2 For every event check all tracks hypotheses combinations - 3^N variants.
- 3 On every step calculate t_0 and χ^2 and find χ_{min}^2 .

$$\chi^2 = \sum^N \frac{(t_0 + tof_{ik} - t_i)^2}{\sigma_t^2 + \sigma_{p_i}^2}, \quad t_0 = \frac{1}{\omega} \sum^N \frac{t_i - tof_{ik}}{\sigma_t^2 + \sigma_{p_i}^2}, \quad \omega = \sum^N \frac{1}{\sigma_t^2 + \sigma_{p_i}^2} \quad (2)$$

- 4 Time complexity $O(N \cdot 3^N)$.

How brute force works:

- 1 Choose particles types to make tof hypotheses $\rightarrow [\pi^\pm, K^\pm, p^\pm]$.

$$tof_{ik} = \frac{L_i}{c} \sqrt{1 + \frac{m_k^2}{p_i^2}} \quad (1)$$

- 2 For every event check all tracks hypotheses combinations - 3^N variants.
- 3 On every step calculate t_0 and χ^2 and find χ_{min}^2 .

$$\chi^2 = \sum^N \frac{(t_0 + tof_{ik} - t_i)^2}{\sigma_t^2 + \sigma_{p_i}^2}, \quad t_0 = \frac{1}{\omega} \sum^N \frac{t_i - tof_{ik}}{\sigma_t^2 + \sigma_{p_i}^2}, \quad \omega = \sum^N \frac{1}{\sigma_t^2 + \sigma_{p_i}^2} \quad (2)$$

- 4 Time complexity $O(N \cdot 3^N)$ - **very slow!!!**

How genetic algorithm works:

- 1 Create population of random candidate solutions - $v([m_i]_k)$.
- 2 Create mutant vector from random candidates in population (DE-inspired):

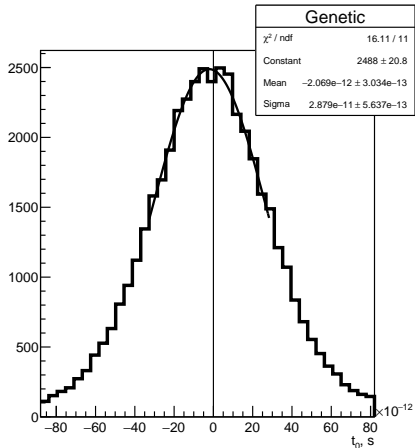
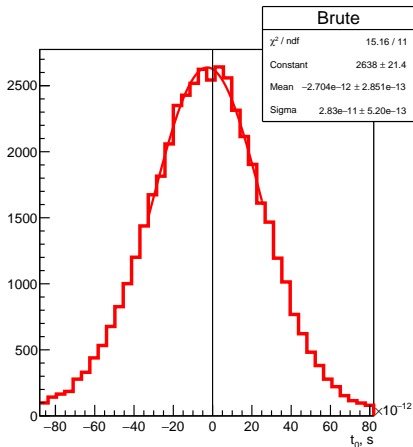
$$v_{mut} = v_r + v_p - v_q \quad (3)$$

- 3 Check if $\chi_{mut}^2 < \chi_r^2$ then replace v_r with v_{mut} . If not, population remains unchanged - **Darwinian selection**.
- 4 Repeat
- 5 After some number of steps stop and choose χ_{min}^2 as an answer.
- 6 Time complexity - $O(N \cdot N_{population} \cdot N_{steps})$, $800 < N_{steps} < 1000$.

Genetic algorithm vs Brute force

- Brute force gives solution with minimal χ^2 , but very slowly.
- Genetic algorithm has less accuracy, but much faster.

Comparison of 2 algorithms was done on events with number of tracks $4 < N < 15$.



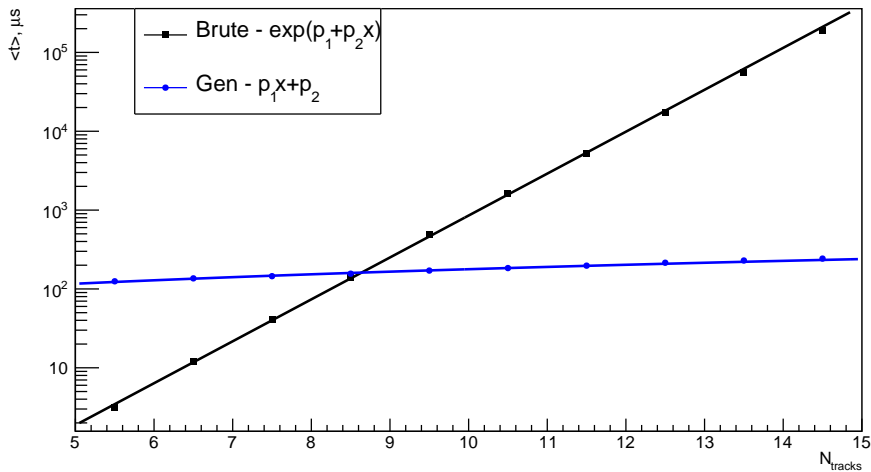
Genetic algorithm vs Brute force

- ① Brute Force resolution of t_0 is 28 ps and Genetic Algorithm is 29 ps.
- ② PID event efficiency for Brute Force is 66.7% and for Genetic algorithm it is 63.3%.
- ③ PID track efficiency for Brute Force is 97.2% and for Genetic algorithm is 96.8%.

PID event efficiency - percentage of events where all tracks were guessed correctly.

PID track efficiency - percentage of tracks that were guessed correctly.

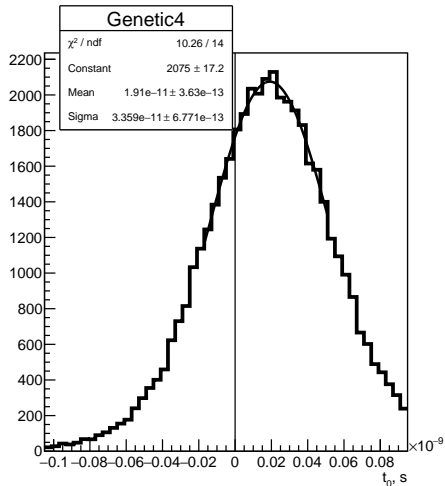
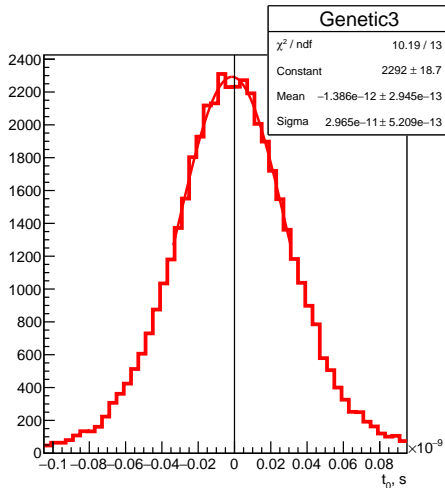
Time complexity

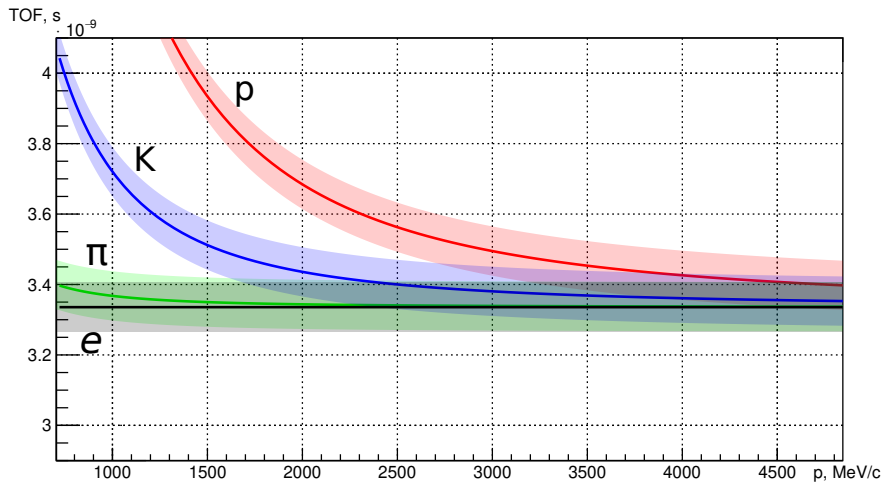


Different hypotheses

Genetic3 hypotheses is $[\pi^\pm, K^\pm, p^\pm]$

Genetic4 hypotheses is $[e^\pm, \pi^\pm, K^\pm, p^\pm]$





Particle Identification

Some possible strategies:

- 1 Take particles types from χ^2 minimum,
- 2 Bayesian approach,
- 3 N-sigma criteria.

In 2 and 3 strategies we exclude particle from determination of t_0 to avoid correlations.

Bayesian approach	N-sigma criteria
$P(H_i \vec{S}) = \frac{P(\vec{S} H_i)C(H_i)}{\sum_{k=\pi,K,p} P(\vec{S} H_k)C(H_k)}$	$n_{\sigma_k}^i = \frac{S_i - \hat{S}_i(m_k)}{\sigma_k^i}$
$P(S H_i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{1}{2}n_{\sigma_i}^2\right)$	If $n_{sigma} < 3$ for certain type,
Probability of being particle of certain type	we associate this type with particle → more than 1 type

https://indico.jinr.ru/event/2788/contributions/15415/attachments/11871/19723/26.01.2022_Ivanov.A.V.pdf

Bayes formula:

$$P(H_i|\vec{S}) = \frac{P(\vec{S}|H_i)C(H_i)}{\sum_{k=\pi,K,p} P(\vec{S}|H_k)C(H_k)} \quad (4)$$

$C(H_i)$ -prior probability, $P(H_i|\vec{S})$ - posterior probability, which is calculated iteratively. On first step $C(H_i)$ are chosen flat.

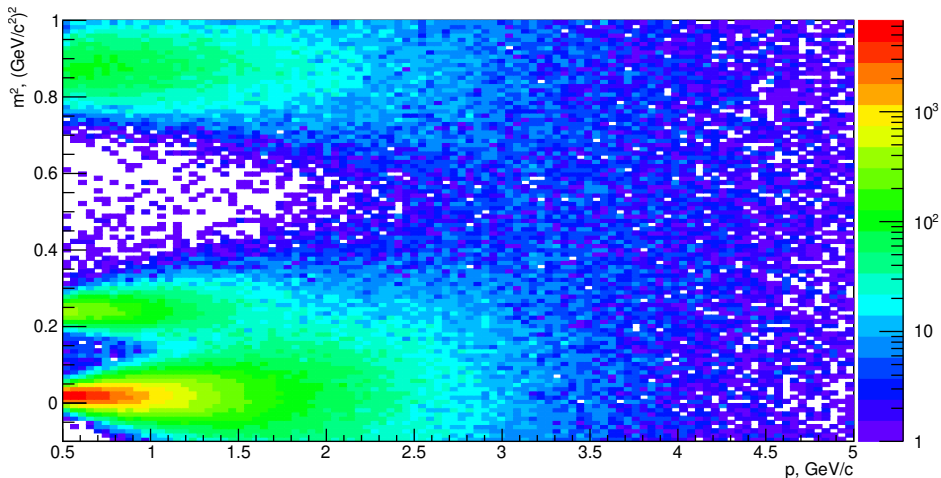
Detector signal:

$$P(S|H_i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{1}{2}n_{\sigma_i}^2\right) \quad (5)$$

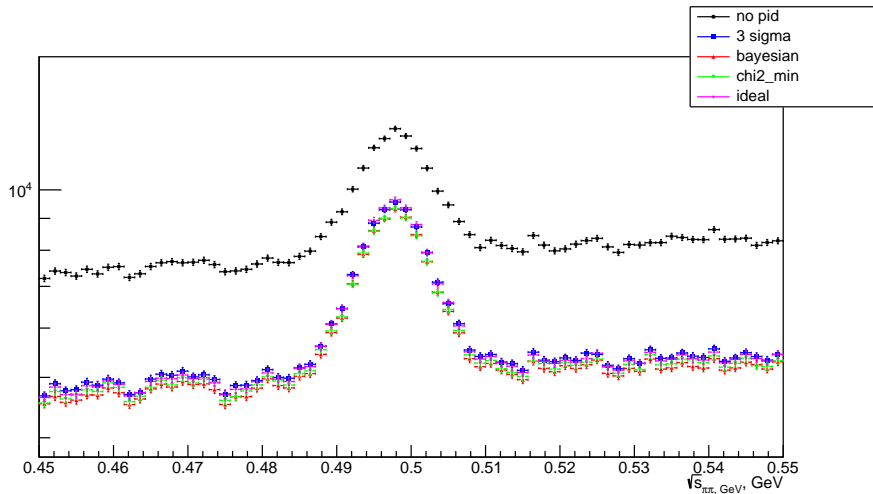
In case of multiple signal(i.e $dE/dx + TOF$):

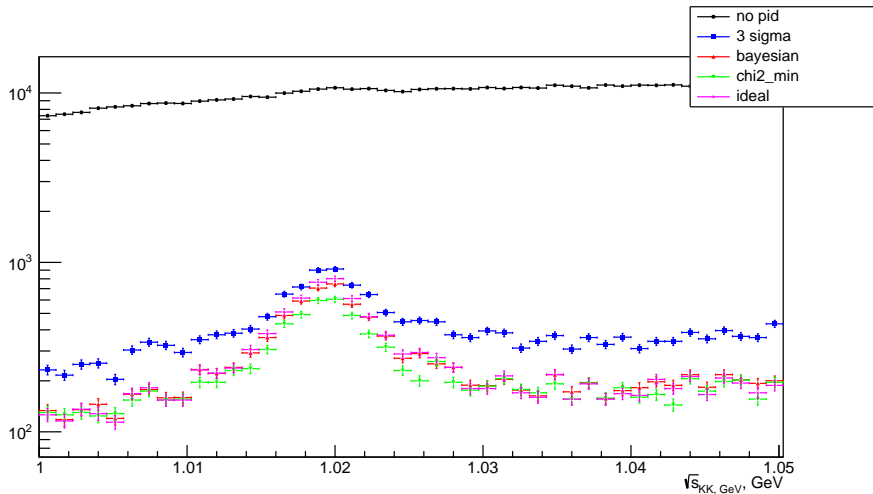
$$P(S|H_i) = \prod_{k=dE/dx, TOF} P(S|H_i)_k \quad (6)$$

PID masses calculated with exclusion

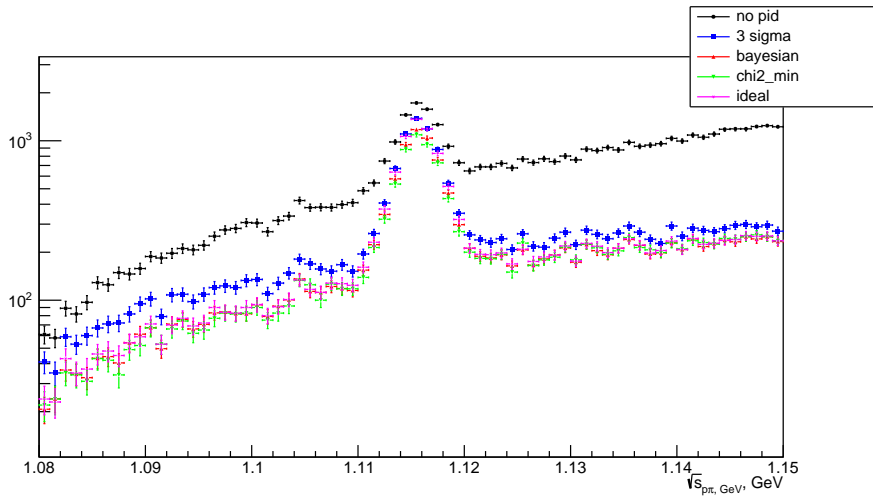


Signals: K_S

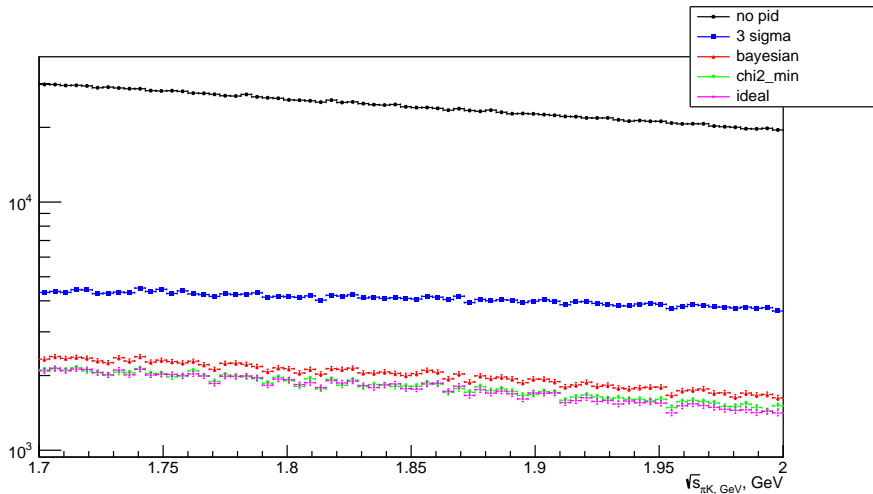




Signals: Λ



Signals: D^0



- ① In events with low multiplicity ($4 < N < 15$) Brute Force on average spend 5 ms per event, and Genetic algorithm 0.160 ms.
- ② Run time grows slowly as function of multiplicity for Genetic algorithm.
- ③ Brute Force resolution of t_0 is 28 ps and Genetic Algorithm is 29 ps.

- 1 Compare DE-inspired algorithm with other types of Genetic algorithms.
- 2 Optimise Genetic algorithm to decrease run time.
- 3 Include t_0 determination algorithm in SPD ROOT.
- 4 PID procedure needs further studies.

Thank you for your attention!