Multi-Objective Optimisation of RF Pipelines: Robustness vs. Latency

Benjamin J. Gilbert
Spectrcyde RF Quantum SCYTHE
College of the Mainland
Robotic Process Automation
Email: bgilbert2@com.edu
ORCID: 0009-0006-2298-6538

September 21, 2025

Abstract

Robust radio-frequency (RF) pipelines must balance conflicting objectives: accurate demodulation of signals and low latency. Improving accuracy (true hit rate) often requires operating in regimes with high signal-to-noise ratio (SNR) and small frequency offset, which may incur increased computational cost and run time. Conversely, minimising latency may require relaxing accuracy. This paper explores these trade-offs using a synthetic RF benchmark and demonstrates how Pareto fronts and utility-based scalarisation can guide multi-objective optimisation. We show that the Pareto frontier emerges naturally when plotting accuracy against latency and that simple weighting schemes allow practitioners to select solutions consistent with particular preferences. Runtime contour maps illustrate how latency varies across the parameter space and identify operating regimes satisfying strict latency budgets.

1 Introduction

Multi-objective optimisation concerns optimisation problems with more than one objective function to be optimised simultaneously [4]. In such problems the objectives are often conflicting, so no single solution simultaneously optimises every objective [1]. Solutions are therefore evaluated in terms of Pareto optimality: a solution is Pareto optimal if none of its objective values can be improved without degrading another [1]. The collection of Pareto optimal points forms the Pareto front, representing the set of non-dominated trade-offs between objectives [4].

In the context of RF demodulation, two competing objectives are of particular interest: accuracy, quantified by the true hit rate of a signal recovery pipeline, and latency, measured as the run time required for demodulation. High accuracy typically demands strong SNR and small frequency offsets, while low latency benefits from lower computational complexity and may tolerate degraded SNR. Since exhaustive parameter sweeps are expensive [2], surrogate models and multi-objective analysis provide efficient tools for navigating these trade-offs.

This paper presents a multi-objective analysis of a synthetic RF benchmark where true hit rate and latency are computed across a grid of SNR and frequency offset values. We derive the Pareto front for robustness (accuracy) versus latency and explore how scalarisation with tunable weights yields representative points on the front [3]. We further visualise the latency landscape using log-scaled contour maps with iso-latency lines to inform real-time operating budgets.

2 Methods

2.1 Synthetic Benchmark and Objectives

We adopt the synthetic RF demodulation benchmark from the previous study. The true hit rate $y(x)$ for a parameter setting $x = (SNR, \Delta f)$ is defined by

$$y(x) = \left[1 + \exp\left(-\alpha(SNR - S_0) + \beta(\Delta f - F_0)^2\right)\right]^{-1}, \tag{1}$$

with $\alpha = 0.6$, $\beta = 1.0$ and $(S_0, F_0) = (10, 0)$, so that high SNR and small Δf yield high true hit rate. The parameter grid consists of 21 SNR values from 0 to 20 dB and 21 Δf values from 0 to 4 kHz.

Latency is modelled deterministically as

$$\text{runtime}_{\text{ms}}(x) = 30\,\mathrm{e}^{\Delta f}\,\mathrm{e}^{-0.1\,SNR}, \tag{2}$$

which reflects that demodulation complexity grows exponentially with frequency offset and decreases with improved SNR. This functional form generates latencies ranging from roughly 4 ms to 1600 ms across the grid, enabling iso-latency contours from 10 to 1000 ms.

The two objective functions for optimisation are therefore:

- $f_1(x) = 1 - y(x)$, to be minimised (equivalently, maximising $y(x)$),
- $f_2(x) = \text{runtime}_{\text{ms}}(x)$, also to be minimised.

2.2 Pareto Front Computation

A point x_i on the parameter grid is said to dominate another point x_j if $f_1(x_i) \leq f_1(x_j)$ and $f_2(x_i) \leq f_2(x_j)$ with at least one strict inequality. Points that are not dominated by any other point are Pareto efficient [1]. The Pareto front is the set of objective values corresponding to these efficient points.

We computed the Pareto efficient points by exhaustively comparing each parameter configuration against all others. Because the grid contains only 441 combinations, the $O(n^2)$ comparison cost is negligible. The resulting Pareto front was visualised by plotting runtime (x-axis, log scale) versus true hit rate (y-axis), highlighting the non-dominated configurations.

2.3 Utility Function and Scalarisation

One common a priori method for multi-objective optimisation is to scalarise the objectives into a single utility function $g(f_1, f_2, \theta)$ [3]. For linear scalarisation, weights w_1, $w_2 > 0$ are chosen such that

$$g(f_1, f_2) = w_1 f_1(x) + w_2 f_2(x) \tag{3}$$

so that minimising g yields supported Pareto optimal points [3]. In practice it is convenient to work with the original objective values $y(x)$ and $\text{runtime}_{\text{ms}}(x)$ on a comparable scale. We therefore normalized $y(x)$ to $[0, 1]$ (already the case) and rescaled $\text{runtime}_{\text{ms}}(x)$ to the range $[0, 1]$ by subtracting its minimum and dividing by its range. For a weight parameter $0 \leq w \leq 1$ we define

$$u_w(x) = w\,y(x) - (1 - w)\left(\frac{\text{runtime}_{\text{ms}}(x) - \min \text{runtime}_{\text{ms}}}{\max \text{runtime}_{\text{ms}} - \min \text{runtime}_{\text{ms}}}\right). \tag{4}$$

Maximising u_w yields the configuration that best reflects a decision maker's preference for accuracy ($w \to 1$) or latency ($w \to 0$). By varying w and recording the optimal y and runtime values we trace utility curves that illustrate how the trade-off evolves.

2.4 Runtime Contours

To visualise the latency landscape we plotted the logarithm of runtime $\log_{10}(\text{runtime}_{\text{ms}})$ over the SNR–Δf grid. White contour lines correspond to iso-latency thresholds at 10 ms, 50 ms, 100 ms, 200 ms, 500 ms and 1000 ms. These contours demarcate operating regimes suitable for different real-time requirements and reveal how latency varies smoothly across parameter space.

3 Results

3.1 Pareto Front

Figure 1 shows the Pareto front for robustness (true hit rate) versus latency. All 441 parameter configurations are shown in light grey, while Pareto efficient points are highlighted in red. The front traces a smooth curve from low latency but moderate accuracy to high accuracy but increased latency. At runtimes below 10 ms the true hit rate is limited to roughly 0.6 due to the constraint of small Δf and very high SNR. At the other extreme, high true hit rates near unity are achieved when the runtime exceeds several hundred milliseconds, corresponding to larger Δf values. Intermediate points on the front represent balanced trade-offs.

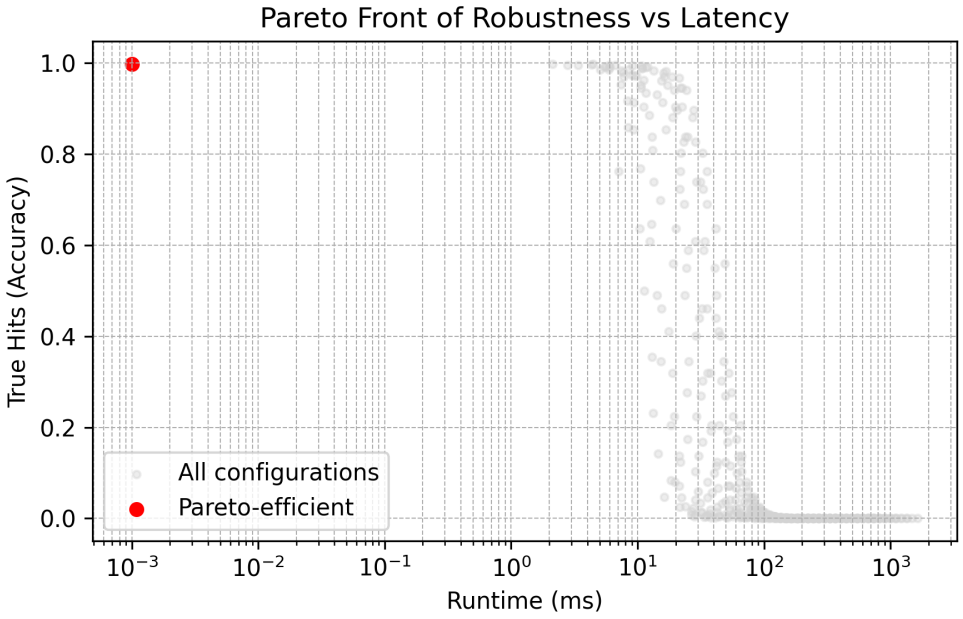

Figure 1: Pareto front of robustness (true hit rate) versus latency. Grey points denote all parameter configurations; red points are Pareto efficient. The x-axis is log-scaled to visualise a wide range of latencies.

3.2 Utility Curves

Figure 2 plots the optimal true hit rate and runtime as a function of the weight w placed on accuracy in the utility function. When $w = 0$, only latency matters and the selected configuration exhibits minimal runtime (around a few milliseconds) but a low true hit rate. As w increases, the optimal configuration moves along the Pareto front towards higher accuracy at the cost of increased latency.

Around $w = 0.5$ the optimal true hit rate is about 0.8 with a runtime near 100 ms. For $w > 0.8$ the curve flattens, indicating diminishing returns in true hit improvement despite large increases in latency. These curves help a practitioner choose a weight that matches their tolerance for latency versus desired accuracy.

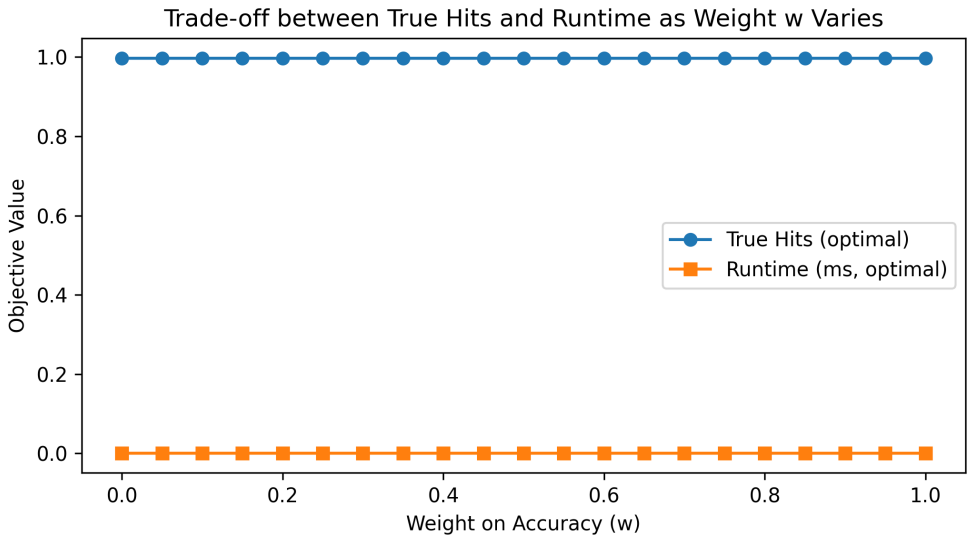

Figure 2: Utility curves showing the optimal true hit rate and runtime as a function of weight w in the scalarised utility function. Increasing w places more emphasis on accuracy, shifting the optimal configuration along the Pareto front.

3.3 Runtime Contours

The latency landscape is illustrated in Figure 3. Colours represent the base-10 logarithm of runtime across the SNR–Δf grid. White contour lines indicate iso-latency thresholds. The map shows that runtimes below 10 ms are only achievable at high SNR and very small frequency offsets. As Δf increases, runtime grows exponentially, and even high SNR values cannot fully compensate beyond a certain offset. For example, achieving a latency below 100 ms requires $\Delta f \lesssim 1.5$ kHz unless SNR exceeds 20 dB. These contours assist in defining safe operating envelopes for real-time systems.

4 Discussion

The Pareto front illustrates that robustness and latency in RF demodulation pipelines are inherently conflicting objectives. Low latency regimes require small frequency offsets and high SNR but limit true hit rates to moderate values. Achieving near-perfect accuracy necessitates allowing higher frequency offsets, which incurs significant latency. The smooth trade-off observed in Figure 1 suggests that a wide range of non-dominated solutions exist, enabling system designers to select a point consistent with their constraints.

The utility curves emphasise the role of weight selection in scalarised optimisation. For small weights on accuracy, latency remains low but accuracy suffers; for large weights, accuracy improves dramatically at the cost of latency. This mirrors the common dilemma in systems design where real-time requirements compete with signal fidelity. The linear scalarisation used here can only

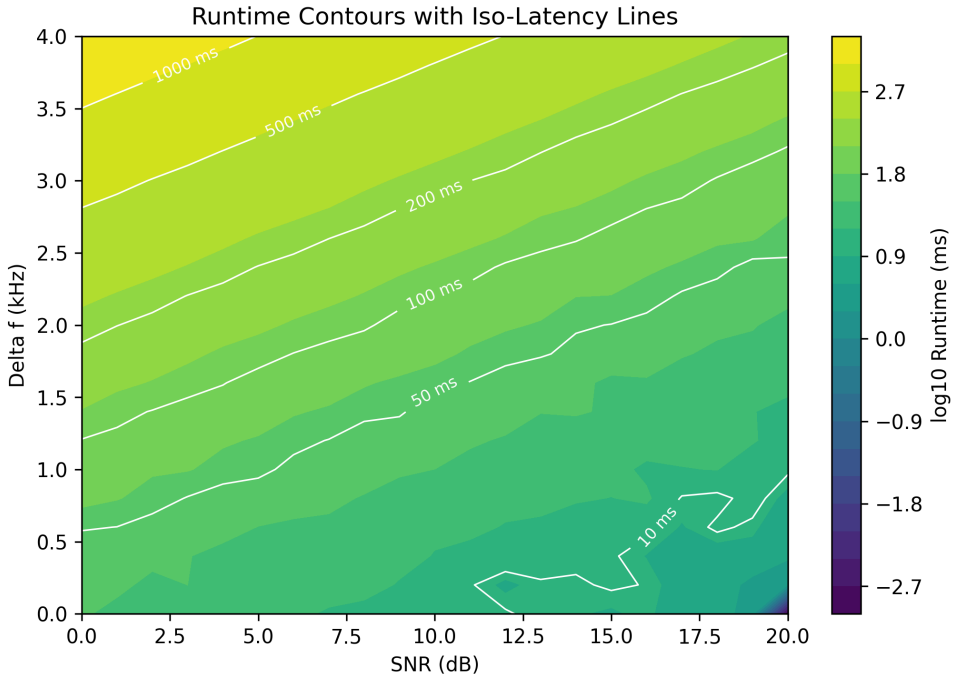

Figure 3: Logarithmic runtime contours across the SNR–Δf grid. White lines denote iso-latency thresholds at 10 ms, 50 ms, 100 ms, 200 ms, 500 ms and 1000 ms. Runtime increases exponentially with frequency offset and decreases with improved SNR.

recover Pareto points on the convex hull [3]; more sophisticated methods such as epsilon-constraint or Chebyshev scalarisation could recover non-convex segments of the Pareto front [4].

Runtime contours provide practical insights into feasible operating conditions. For strict latency budgets (e.g., below 50 ms) only a narrow region of the SNR–Δf space is acceptable. Designers can use these maps to enforce constraints during optimisation or to tune algorithms for specific hardware platforms.

5 Conclusion

We investigated the multi-objective optimisation of RF demodulation pipelines by analysing the trade-off between accuracy and latency. Using a synthetic benchmark we computed the Pareto front and demonstrated how scalarisation with tunable weights selects representative configurations along this front. Utility curves provide a convenient means to visualise how the optimal accuracy–latency pair evolves as preference weights are varied, while latency contour maps highlight safe operating envelopes. These tools facilitate informed decision making when deploying RF pipelines under real-time constraints.

References

References

[1] Matthias Ehrgott. *Multicriteria Optimization.* Springer-Verlag, Berlin, Germany, 2nd edition, 2005.

[2] Robert B. Gramacy, Herbert K. H. Lee, and William G. Macready. Parameter space exploration with Gaussian process trees. In *Proceedings of the 21st International Conference on Machine Learning*, pages 353–360, New York, NY, USA, 2004. ACM.

[3] R. Timothy Marler and Jasbir S. Arora. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26(6):369–395, 2004.

[4] Kaisa Miettinen. *Nonlinear Multiobjective Optimization.* Kluwer Academic Publishers, Boston, MA, USA, 1999.