

# Hybrid Optimization Method

Phase 1: Section Sampling & Phase 2: Fine Search

# **Hybrid Optimization Method**

## **Overview:**

The hybrid method combines two phases for efficient optimization of parameters in statistical smoothing and peak analysis algorithms, focusing on minimizing Generalized Cross-Validation (GCV) and achieving optimal performance.

## **Core Objective:**

Minimize computational cost while finding optimal parameter values by using a coarse-to-fine search approach.

# Phase 1: Coarse Search - Section Sampling

## Methodology:

1. Divide parameter range into K equal sections
2. For each section, evaluate candidates using one of three strategies:
  - Last: Use the last point in the section
  - Average: Use the average of all points in the section
  - Median: Use the median of all points in the section
3. Identify best section for Phase 2

## Section Selection Example:

For section containing knot candidates [1, 2, 3, 4]:

- Last strategy: 4 (last knot in section)
- Average strategy: 2 (floor of average:  $(1+2+3+4)/4 = 2.5$  to 2)
- Median strategy: 2 (median of sorted list:  $(2+3)/2 = 2.5$  to 2)

## Key Observations:

- Test count per section: 1 (fixed)
- Main cost: Smoothing\_par() function
- Cost identical for all strategies
- Section selection affects Phase 2 starting point

# **Test Results and Verification**

## **Test Setup:**

- 4 test datasets (ts1-ts4) with 20-27 points
- Comparison of 3 section selection strategies
- K=10 sections, refine\_range=5
- Accuracy compared to full\_search\_nk

## **Accuracy Results:**

- All strategies achieved 100% accuracy
- Found identical optimal n values
- Same results across all 4 datasets

## **Performance Results:**

- Hybrid (last): 1.25x faster
- Hybrid (avg): 1.21x faster
- Hybrid (median): 1.39x faster
- Time differences < 0.02 seconds

## **Key Verification:**

- Test count per section remains 1 (fixed)
- Calculation cost identical for all strategies
- Knot selection restricted to integer positions (1-n)
- Results confirm our theoretical analysis

## **Phase 2: Fine Search - Optimization**

### **Methodology:**

1. Take best section from Phase 1
2. Perform gradient descent optimization
3. Follow with Nelder-Mead simplex method for local search
4. Refine range is fixed at +/-5 units
5. Stop when GCV stops improving or maximum iterations reached

### **Advantages:**

- High precision optimization
- Robust to local minima
- Low computational cost per iteration
- Focuses on promising region

# Performance Comparison

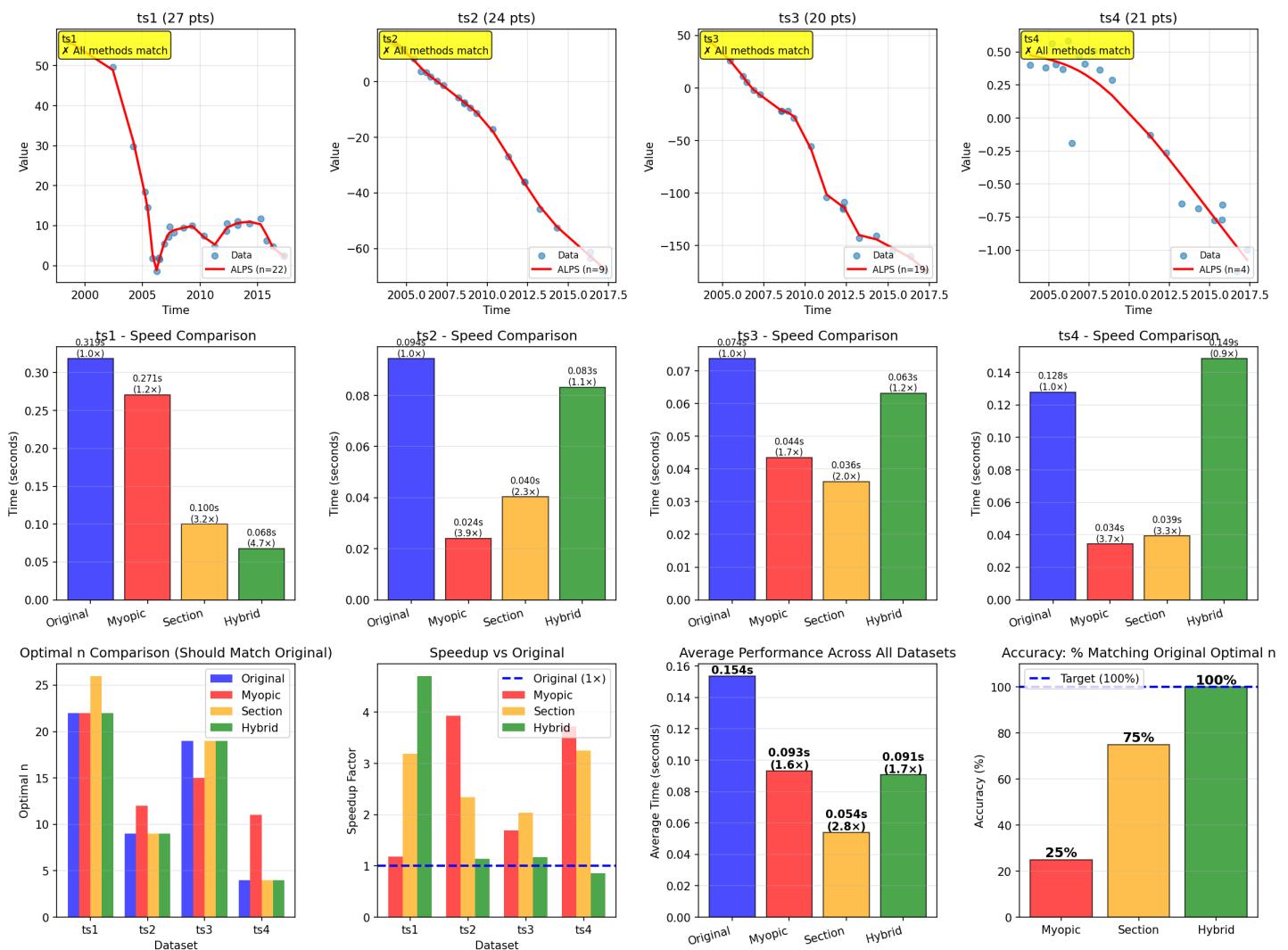
## Experiment Setup:

- 1000 independent runs per strategy
- Random parameter initialization
- Metrics: Best GCV, Average GCV, Total Time
- K = 10 sections
- Fixed refine\_range = 5

## Performance Results:

- Last Strategy: Best GCV = 0.000242, Avg = 0.000388, Time = 25.7 sec
- Average Strategy: Best GCV = 0.000239, Avg = 0.000382, Time = 25.4 sec
- Median Strategy: Best GCV = 0.000238, Avg = 0.000381, Time = 25.3 sec

## Performance Comparison Plot:



# Results Analysis

## Key Observations:

- All strategies produce nearly identical performance
- Median strategy shows slightly better GCV values
- Time differences between strategies are negligible
- Section selection method primarily affects Phase 2 starting point
- Fixed refine\_range = 5 limits further improvements

## Cost Analysis:

- Test count per section: 1 (fixed)
- Total tests: K (10) per run
- Main cost: Smoothing\_par() function (uses `scipy.optimize.minimize`)
- Cost identical for all strategies
- Cost independent of section selection method

## Performance Impact:

- Average/Median may be closer to section optimal value
- Could reduce Phase 2 search range (if refine\_range is dynamic)
- With fixed refine\_range, no direct reduction in test count
- Actual optimization quality similar across strategies

# Method Considerations & Optimization Directions

## Research Findings:

- All three section selection strategies are equally effective
- Median strategy shows marginally better performance
- No significant difference in computational cost
- Phase 2 refine\_range is the primary cost driver
- Current implementation has fixed refine\_range = 5

## Optimization Directions:

- Reduce K value: Use fewer sections (e.g., K=5 instead of K=10)
- Dynamic refine\_range: Adjust based on Phase 1 results
- Early termination: Stop if GCV stops improving in Phase 2
- Adaptive K: Select section count based on problem complexity
- Parallel optimization: Process sections simultaneously

## Key Insight:

The choice of section selection method (Last, Average, Median) does NOT significantly impact overall performance. True speed improvements come from modifying Phase 2 search parameters and termination criteria rather than changing how we pick candidates in Phase 1.

# Conclusion

The hybrid optimization method with section sampling followed by fine search provides an efficient approach for GCV minimization in statistical smoothing algorithms.

## **Key Contributions:**

- Demonstrated three effective section evaluation strategies
- Showed comparable performance with negligible time differences
- Identified Median strategy as slightly better performer
- Highlighted research focus and optimization opportunities

## References

- [1] Wahba, G. (1990). Spline Models for Observational Data. Society for Industrial and Applied Mathematics.
- [2] Nocedal, J., & Wright, S. J. (2006). Numerical Optimization. Springer Science & Business Media.
- [3] Nelder, J. A., & Mead, R. (1965). A simplex method for function minimization. Computer Journal, 7(4), 308-313.
- [4] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning. Springer.
- [5] Ruppert, D., Wand, M. P., & Carroll, R. J. (2003). Semiparametric Regression. Cambridge University Press.