1. Introduction and Motivation
   1. Overview
   2. General problem: Slow solvers in dynamic environments
   3. Typical tradeoff: Rapid suboptimal solution or slow optimal solution
   4. Approach
2. Related Work
   1. CBR
   2. Planning
   3. Classification
   4. Statistical Sampling
3. Problem Space Analysis
   1. Representation
      1. Simple
      2. Hypersurface
      3. Tree
   2. Memory Requirements
   3. Time Requirements
4. Problem Space Analysis Approximation
   1. Sampling-Classification
   2. Sampling-Classification + Bias
   3. Solution Border Estimation
   4. Sampling-Classification + Active Learning
   5. Select from Sampled Solutions
   6. Support Vector Machine
   7. Support Vector Machine + Solution Border Estimation
   8. Algorithm Complexity
5. Evaluation: Test Domains
   1. Domains
      1. Dynamic Traveling Salesman Problem
      2. Dynamic 0-1 Knapsack Problem
      3. Elevator Domain
   2. Results
      1. Benchmarks
      2. Time results
      3. Match Accuracy
      4. Utility Loss from Optimal
         1. Vs from scratch
         2. Vs plan repair
      5. Time
   3. Analysis
      1. SC is reasonable approach
      2. SC+Bias is does not deliver consistent results
      3. SBE is very good, but only works in two dimensions
      4. SC+AL
      5. SSS works pretty well, dependent on solution discovery
      6. SVM as standalone does not work very well
      7. SVM + SBE works well
6. ~~Approximation Algorithm Configuration~~
   1. ~~Contract Algorithm~~
   2. ~~Real-time Algorithm~~
   3. ~~(sample rate, alpha, polling radius) = f (time, space, accuracy threshold)~~
7. ~~Applications~~
   1. ~~UAV Allocation~~
8. Future Work
   1. Solution-Problem-Utility Map Approximation
   2. Problem Decomposition
   3. Meta-Algorithm: Robot Transmission
   4. WSN
   5. Increasing time & space efficiency
9. Appendix
   1. Framework
      1. Problem Instance Definition
      2. Solution Definition
      3. Solver
      4. Visualizer