

I tested the algorithms using three primary domains. The traveling salesman problem (TSP), knapsack problem, and an elevator problem. The TSP and knapsack problem are classic problems in optimization and computer science. The elevator problem is a challenge domain created for the AAAI International Planning Competition (IPC). All the problems are NP-complete or NP-hard, and quickly become intractable with complex enough problem instances.

I used the TSP as an initial domain for algorithm validation and development. I generated problem instances ranging from 5 cities to 100 cities, representing a range of problem complexity. The algorithms were developed based upon the insights from tests of each preceeding algorithm, which will be reflected in some of the discussion. The initial algorithm, sample-classification (SC), solves a random sample of the problem instances and uses classification based on nearest neighbor to assign solutions discovered during the initial sample to each unsolved problem instance. In the initial experiment, all the solutions of solved problem instances within a static radius of an unsolved problem instance were polled and the solution with the plurality was assigned to the unsolved problem instance. Subsequent experiments weighted the solutions by the reciprocal of the distance or the distance squared, this giving more weight to the solutions of problem instances closer to the unsolved instance. Another approach uses an expanding radius to consider additional solutions if there did not exist a plurality. The results of the SC experiments are in figure X.

Based on the results of the SC experiments, it became apparent that the larger solution regions tended to be represented in the approximated PS Map, but smaller regions tended to disappear. This occurred due to the lower probability of the initial random sample choosing a problem instance in that region, resulting in either a particular region not being represented in discovered solutions or that other, more probabilistic, solutions being assigned. One attempt to mitigate this effect was inspired by observing that within

high-quality PS Map, it appears that more rapid changes in solutions and smaller solution regions tend to exist near city locations. The SC+bias algorithm attempts to take advantage of this observation by biasing samples in the regions near cities. The city radius and bias parameters determine, respectively, the radius of the region around a city to apply the bias and how much to bias the samples. The results of this experiment are in figure X. The results do not show a clear means of applying the parameters to achieve consistently good results.

Sampling classification with active learning (SC+AL) is another attempt to allow for smaller solution regions to be approximated effectively. SC+AL may be considered a generalization of SC+bias in that it allows more concentrated sampling in regions of the problem space in which the classification appears ambiguous. For example, if two solutions are both strong candidates to be assigned to a specific problem instance, then SC+AL would allow the problem instance to be solved rather than risk assigning an incorrect solution. Similarly, if there are no strong candidates for a particular problem instance, then SC+AL would allow the problem instance to be solved rather than assign an arbitrary solution to it. The results of this experiment are shown in figure X.

The solution border estimation algorithm (SBE) considers the mathematical features of the TSP. It calculates the border by recognizing that the border between any two solutions is represented by equating the distance functions of the two solutions. Unfortunately, at the time of this experiment, I did not find a Java library that could solve the complex equations that resulted from this technique. The SBE-trace technique is inspired by SBE, however it finds borders between two solutions by searching the space between two problem instances with known solutions. Thus, a binary search can be employed. Assuming that the border between two solutions is continuous, then the remainder of the border can be found by comparing the utility of the two solutions at each problem instance. The results of SBE-trace are shown in figure X. However, SBE-trace is only suitable for two dimensional PS Map

approximation, and thus I did not emphasize this algorithm in the subsequent experiments.

The support vector machine algorithm (SVM) uses a support vector machine to try to generalize the idea of SBE to multiple dimensions. Support vector machines calculate a maximum margin plane to separate different classes. The observations in this application are the sampled problem instances labeled with their solutions. The results of this approach are in figure X.

One disadvantage of the SVM-based approach is that it can misclassify problem instances. SVM determines the borders between two solution regions by creating a margin as far as possible between known solutions instances. This results in a border that is approximately midway between known solutions. This has been shown to be a good optimization technique in general, however, it does lead to misclassifications when the actual border does not conform to this approximation. By applying additional samples in key locations, the bounds of the margins calculated by the SVM can be made tighter and thus more consistent with the actual borders. In this approach, the first step is an initial set problem instances are sampled and solved. The second step applies the binary search used in the SBE-trace algorithm to each distinct pair of solutions. This results in problem instances that represent solutions that are on the border between the distinct pair of solutions. Finally, those problem instances and the labeled solutions are added to the training set for the SVM. The results of this approach are in figure X.

The knapsack domain demonstrates the applicability of the algorithms in a different domain. One difference of this domain versus the TSP is that it entails more abstract representation of distance, as an item's weight and value characteristics do not directly correspond to location and distance as directly as they do for the cities within the TSP domain. The high-quality solution PS Map's characteristics also differ in this domain TSP. For example, looking at the high-quality PS Map, one can see that, whereas the TSP domain had very circular homogeneous regions, the knapsack domain has rectangular homogeneous

regions. I apply the same algorithms to this domain, with the exception of the SBE-trace algorithm which is only suitable for problem spaces of two dimensions.

Finally, the elevator domain represents a more traditional planning domain. The TSP and knapsack domains can be considered optimization problems as well as planning problems. The elevator domain falls into the more traditional realm of planning where one has to find steps to accomplish a goal - there is no direct mathematical representation of the domain. Also, this domain is expected to be more challenging for the algorithms because the homogeneous regions are likely to be smaller and less regular. Lastly, this domain represents a another level of abstraction in that the solutions that are applied to the domain are not necessarily those that the algorithms will operate upon. I also apply the algorithms to this domain, again with the exception of SBE-trace.