CHAPTER V

EXTRACTION OF THE CORRECT ANSWER

Introduction and Overview

Once the T_list discussed in Chapter IV is formed, the next step is to extract the proper answer from it. Winter described this in step 7 of his algorithm (Figure 17). His description stated that unions of FST's saved in T_list were to be formed subject to constraints described in his paper. The shortest union is the SMT for the original points.

The constraints he described were quite obvious considering the definition of an SMT. First, the answer had to cover all the original points. Second, the union of FST's could not contain a cycle. Third, the answer is bounded in length by the length of the MST.

This led Winter to implement a simple exhaustive search algorithm over the FST's in T_list. This approach yields a sample space of size $\mathcal{O}(2^m)$ (where m is the number of trees in T_list) that has to be searched. This exponentiality is born out in his work where he stated that for problems with more than 15 points "the computation time needed to form the union of FST's dominates the computation time needed for the construction of the FST's [36]." An example of the input the last step of Winter's algorithm receives (T_list) is given in Figure 23. The answer it extracts (the SMT) is shown in Figure 24.

Incompatibility Matrix

Although the SMT problem attracted many people through the early 70's the lack of results seemed to dampen most of their spirits. It wasn't until Winter's algorithm was published that the research community received the spark it needed to work on the SMT problem with renewed vigor. With the insight Winter provided into the problem, an attempt to computerize the solution of the SMT problem began anew.

Enhancement of this algorithm was first attempted by Cockayne and Hewgill at the University of Victoria. For this implementation Cockayne and Hewgill spent most of their work on the back end of the problem, or the extraction from T_list, and used Winter's algorithm to generate T_list. This work on the extraction focused on what they termed an

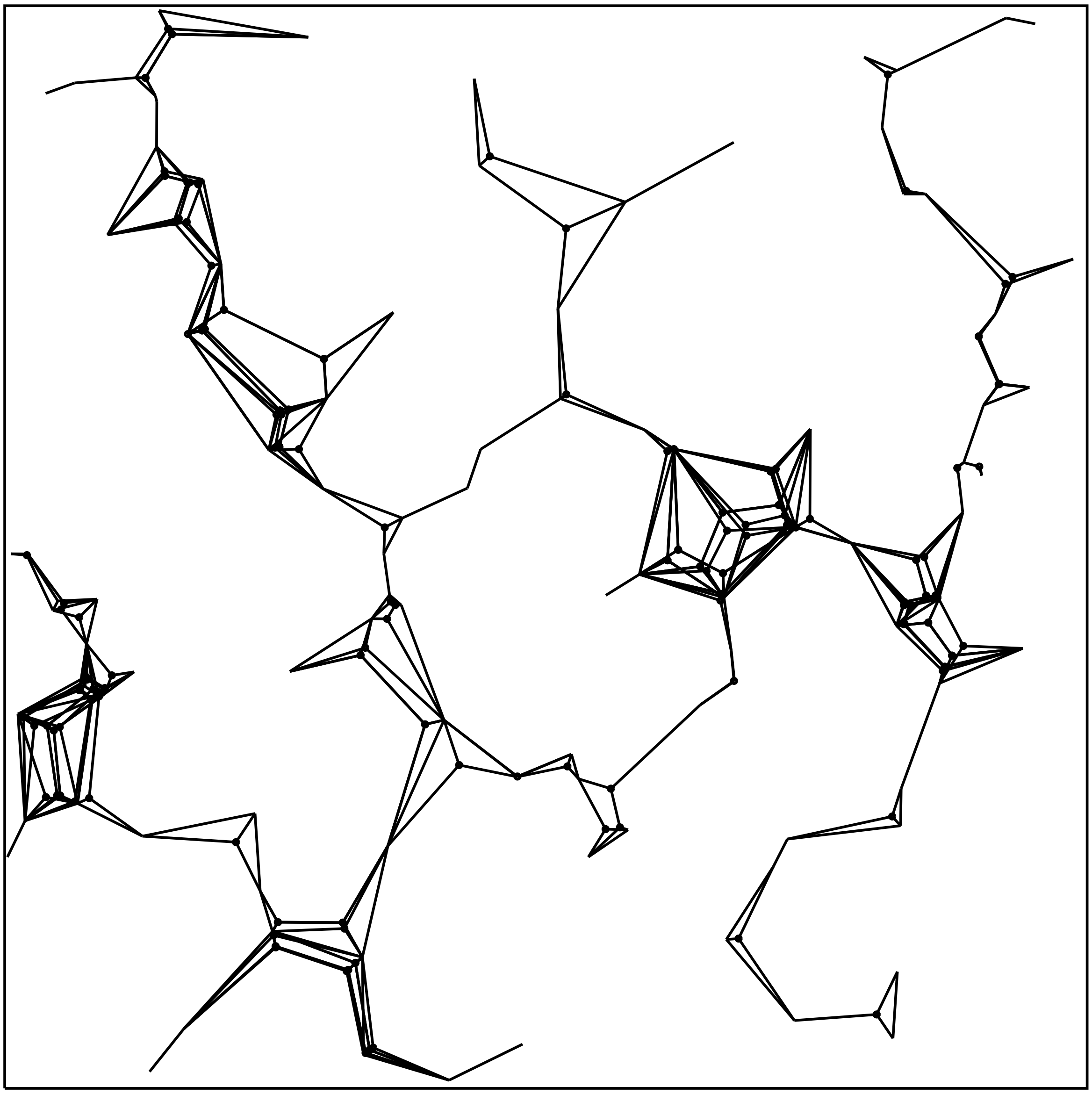

Figure 23. T_list for a random set of points.

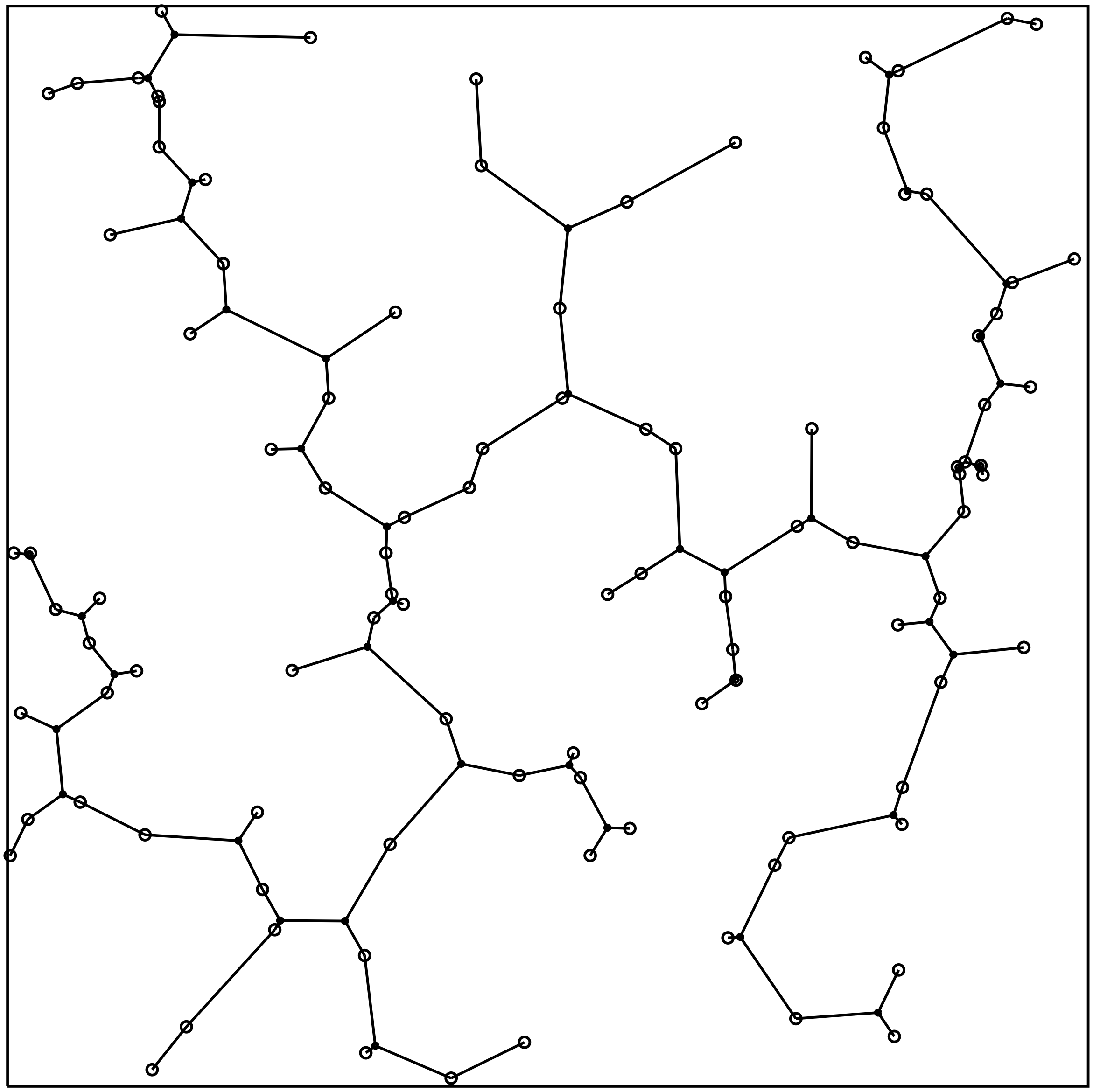

Figure 24. SMT extracted from T_list for a random set of points.

incompatibility matrix. This matrix had one row and one column for each member of T_list. The entries in this matrix were flags corresponding to one of three possibilities: *compatible*, *incompatible*, or *don't know*. The rationale behind the construction of this matrix is the fact that it is faster to look up the value in a matrix than it is to check for the creation of cycles and improper angles during the union of FST's.

The first value calculations for this matrix were straightforward. If two trees do not have any points in common then we *don't know* if they are incompatible or not. If they have two or more points in common then they form a cycle and are *incompatible*. If they have only one point in common and the angle at the intersection point is less than 120° then they are also *incompatible*. In all other cases they are *compatible*.

This simple enhancement to the extraction process enabled Cockayne and Hewgill to solve all randomly generated problems of size up to 17 vertices in a little over three minutes [38].

Decomposition

The next focus of Cockayne and Hewgill's work was in the area of the decomposition of the problem. As was discussed earlier in Chapter III, the best theorems for any problem, especially non-polynomial problems, are those of the form "If property $\mathcal{P}$ exists then the problem can be decomposed." Since the formation of unions of FST's is exponential in nature any theorem of this type is important.

Cockayne and Hewgill's theorem states: "Let A_1 and A_2 be subsets of A satisfying (i) $A_1 \bigcup A_2 = A$ (ii) $|A_1 \bigcap A_2| = 1$ and (iii) the leaf set of each FST in T_list is entirely contained in A_1 or A_2. Then any SMT on A is the union of separate SMT's on A_1 and A_2 [38]." This means that if you break T_list into biconnected components, the SMT will be the union of the SMT's on those components.

Their next decomposition theorem allowed further improvements in the calculation of SMT's. This theorem stated that if you had a component of T_list left from the previous theorem and if the T_list members of that component form a cycle, then it might be possible to break that cycle and apply the previous algorithm again. The cycle could be broken if there existed a vertex v whose removal would change that component from one biconnected component to more than one.

With these two decomposition theorems, Cockayne and Hewgill were able to calculate the SMT for 79 of 100 randomly generated 30 point problems. The remaining 21 would not decompose into blocks of size 17 or smaller, and thus would have taken too much computation time [38]. This calculation was implemented in the program they called EDSTEINER86.

Forest Management

Their next work, which was implemented in a program called EDSTEINER89, focused on improvements to the *incompatibility matrix* previously described. Their goal was to reduce the number of *don't know*'s in the matrix and possibly remove some FST's from T_list altogether.

They proposed two refinements for calculating the entry into the *incompatibility matrix* and one Tree Deletion Theorem. The Tree Deletion Theorem stated that if there exists an FST in T_list that is incompatible with all FST's containing a certain point a then the original FST can be deleted since at least one FST containing a will be in the SMT.

This simple change allowed Cockayne and Hewgill to calculate the SMT for 77 of 100 randomly generated 100 point problems [25]. The other 23 problems could not be calculated in less than 12 hours and were therefore terminated. For those that did complete, the computation time to generate T_list had become the dominate factor in the overall computation time.

So the pendulum had swung back from the extraction of the correct answer from T_list to the generation of T_list dominating the computation time. In Chapter VI we will look at the results of the parallel algorithm presented in Chapter IV to see if the pendulum can be pushed back the other way one more time.