CSI 5165 Combinatorial Algorithms | Fall 2010
Computer Science | University of Ottawa

Homework Assignment #2 (100 points, weight 15%)
Due: November 10 (in lecture)

---

1. (Exercise 5.1 in the textbook) (15 marks) Determine the sizes of the following neighbourhoods in the associated optimization problems:
   - an exchange neighbourhood for the **Uniform Graph Partition** problem
   - a 2-opt neighbourhood for the **travelling salesman** problem.
   - a neighbourhood N_{d_0} for a problem in which the universe consists of all permutations of an n-set.
   - a neighbourhood N_{d_0} for a problem in which the universe consists of all $(0,1)$-vectors of length n having weight w.

2. (35 marks) Consider the algorithm you designed in assignment#1 to generate all Steiner triple systems of order n.
   - Use Knuth's method to estimate the backtracking tree size for the cases of $n = 7, 9, 13$. Please, for each n, provide some analysis by varying the number of probes used, and showing the various estimations obtained. Note that for $n = 7, 9$ you are able to compare the estimation with the actual tree size you measure in last assignment (please include the actual tree size).
   - In this part, you will use a more general version of the theorem seen in class (Knuth 1975): associate with each node X in the backtracking tree, a weight $w(X)$. The goal is to estimate $w(T) = \sum_{X \in T} w(X)$. The estimate W obtained from the path $X_0, X_1, \ldots, X_k$, following the same notation as in the class notes, is:

     $$W = w(X_0) + |\mathcal{C}_0|w(X_1) + |\mathcal{C}_0||\mathcal{C}_1|w(X_2) + \cdots + |\mathcal{C}_0||\mathcal{C}_1|\cdots|\mathcal{C}_{k-1}|w(X_k).$$

     The revised theorem states that the expected value of W returned by the algorithm is $w(T)$. The previously seen theorem is equivalent to $w(X) = 1$ for all nodes X in the backtracking tree T.
     Use a method based on this generalized theorem to estimate the **number** of Steiner triple systems of order n; in particular, give the weight function used. Do a similar analysis as the previous case and compare it with the actual values, which are known for $n = 7, 9, 13$. The following web page the number of such systems for $n = 1, 3, 7, 9, 13, 15, 19$:

     `http://www.research.att.com/~njas/sequences/A001201`

3. (50 marks) (Exercise 5.3 in the textbook) For sparse graphs, it has been shown that Algorithm 5.9 does not perform very well when solving the **Uniform Graph Partition** problem. Give pseudocodes and implementations for **two out of the following three** types of methods for solving the **Uniform Graph Partition** problem, assuming that the underlying graph is a sparse graph:
   - a simulated annealing algorithm
   - a tabu search algorithm
   - a genetic algorithm

   For each method, explain details of the method such as the neighbourhood function, neighbourhood search, recombination methods, etc.
   Compare the effectiveness of your algorithms by running them on random sparse graphs. Do they perform better than Algorithm 5.9? (note: one convenient way to generate a random sparse graph is to use Algorithm 4.21 with small values of δ). You may use the C implementations of the algorithms in the textbook (Algorithm 4.21, 5.9), available from the textbook web page:

   `http://www.math.mtu.edu/~kreher/cages/Src.html`