Thomas Jansen
University College Cork
October 16th, 2012

Assignment 8

Topic 'Applying Randomised Search Heuristics'

In the following it is assumed that you know what a connected component of an undirected, weighted graph is. An explanation can be found at the bottom of this sheet.

We want to solve the following optimisation problem. We are given an undirected, weighted graph (G, w) where $G = (V, E)$ with $E = \{e_1, e_2, \ldots, e_n\}$ and $w \colon E \to \mathbb{R}^+$. We are looking for a subset of edges $E' \subseteq E$ such that the graph (V, E') has exactly the same number of connected components as G. Among all these subsets we are looking for one that has minimal total edge weight. Formally speaking, we want to minimise $\sum_{e \in E'} w(e)$ or, equivalently, maximise $-\sum_{e \in E'} w(e)$.

We want to apply some standard randomised search heuristic (like random local search or an evolutionary algorithm) and use the search space $S = \{0,1\}^n$. We define a mapping m by $m(s) = \{e_i \mid s[i] = 1\}$.

Problem 1 If $(V, m(s))$ has the same number of connected components as G we have $-\sum_{e \in E'} w(e)$ as function value. (Remember that we maximise.) If $(V, m(s))$ has a different number of function values we need to define a value. Let us define the value for such search points as $-\infty$.

Prove a lower bound on the expected optimisation time of the (1+1) EA for the following inputs:

- $V = \{v_1, v_2, \ldots, v_n, v_{n+1}\}$
- $E = \{\{v_1, v_2\}, \{v_2, v_3\}, \{v_3, v_4\}, \ldots, \{v_{n-1}, v_n\}, \{v_n, v_{n+1}\}\}$
- $w(e) = 1$ for all $e \in E$

Problem 2 We now use a different method of defining a function value for those s where it is not defined for $m(s)$. Let c be the number of connected components in G. Let c_s be the number of connected components in $(V, m(s))$. Let $W = 1 + \sum_{e \in E} w(e)$. We define $f(m(s)) = -(c_s + 1 - c) \cdot W$ for such $m(s)$.

Prove an upper bound on the expected number of steps the (1+1) EA needs to find some graph with c connected components.

Problem 3 <u>**Implementation Task**</u>
Implement the fitness function from Problem 2.

For $n \in \{10, 20, 30, \ldots, 100\}$ create random graphs with n edges in the following way, one graph for each value of n (resulting in 10 random graphs). The graph for n has $\lfloor \sqrt{n} \ln(n) \rfloor$ nodes. Generate edges randomly by picking two nodes uniformly at random and insert that edge. Note that each pair of nodes cannot have more than one edge. Assign weights to the edge by selecting the weights from $\{1, 2, 3, \ldots, \lceil \sqrt{n} \rceil\}$ uniformly at random, independently for each edge.

For each graph, perform 30 runs for random local search and the (1+1) EA, both for n^3 function evaluations. Plot a graph of the average function values obtained for both algorithms. Can you explain your results?

Assignments are handed out on each Friday during the lecture. Written solutions are due the next Wednesday. Feedback is given and solutions are discussed the Wednesday after that.

Consider an undirected graph $G = (V, E)$. It is not important if the graph is weighted or not. In the following we can and will ignore the edge weights.

We say that two nodes $v_1, v_2 \in V$ are connected if there is a sequence of edges leading from v_1 to v_2. We say that a set of nodes $V' \subseteq V$ is a connected component if

1. for any two nodes $v_1, v_2 \in V'$ the two nodes are connected, and
2. for any two nodes $v_1 \in V'$ and $v_2 \notin V'$ the two nodes are not connected.