Subset sum problem

Thomas Jager Torbjørn Skagestad Kristian Tveiten

October 18, 2007

Abstract

In this project we want to find out if Learning Automata is suitable for solving the Subset Sum Problem.

1 Project Description

In this project we want to find out if Learning Automata is suitable for solving the Subset Sum Problem. And compare performance with other ways of solving the problem will be an important part of the project.

2 Subset sum problem

Given a set of integers, does the sum of some non-empty subset equal to exactly zero. For example, given the set { -7, -3, -2, 5, 8}, the answer is yes because the subset {-3, -2, 5} sums to zero. The exact solution to the subset sum problem is NP-Complete, and is perhaps the simplest of the NP-Complete problems to describe.

$S$is a solution.
$A = \{x_1, x_2, ..., x_n\}$
$S \neq \oslash$
$\sum S = 0$

2.1 NP-Complete

NP-Complete problems are a class of complexity problems that run in polynomial time. Polynomial time refers to the computation time of a problem where the run time, m(n), is no greater than a polynomial function of the problem size, n. Written mathematically using big O notation, this states that $m(n) = O(nk)$ where k is some constant that may depend on the problem. For example, the quicksort sorting algorithm on n integers performs at most An^2 (Worst case) operations for some constant A. Thus it runs in time $O(n^2)$ and is a polynomial time algorithm.

2.2 Approximate Solution

Some problems don't need a exact solution. Approximate solutions are often useful in real life situations like optimizing packing of a container. In these situations it is desirable to get a sum as close to the exact solution as possible. There are many things to consider when searching for a approximate solution. If no exact solution exist in the subset, we will need to set a threshold for how many runs should be done. A simple way to implement this is to keep the sum closest to the exact solution, and discard all other. When the number of runs set as a threshold then is completed, you chose the subset that offers the solution closest to the exact solution. This type of solution is applicable to the knapsack problem for instance.

2.3 Exact Solution

In cryptography the subset sum problem comes up when breaking encryption keys. When the attacker knows the message and the cipher text (Known plaintext attack), finding the key boils down to a subset sum problem. A key that is not equal to the real key is useless and therefore an exact solution is needed.

3 Project Goals

We want to implement a learning automata solver for both approximate solution and exact solution. We will also implement some known algorithms for solving the problem to compare speed. We also want to compare diffrent reward/punishment strategies aswell as decision strategies.