IOSR Journal of Mathematics (IOSR-JM)
e-ISSN: 2278-5728, p-ISSN: 2319-765X. Volume 12, Issue 5 Ver. III (Sep. - Oct.2016), PP 54-57
www.iosrjournals.org

The optimal solution for the Knapsack problem find by The Electronic algorithm

M. Sghiar
9 Allée capitaine J.B. Bossu, 21240, Talant, France

Abstract: *In [8], the electronic algorithm has allowed to find the optimal solutions for the travelling salesman problem. In this article, I demonstrate that the same algorithm will allow to find the optimal solutions for the knapsack problem.*
Keywords: *Graph, Hamilton cycles, P=NP, the knapsack problem, the travelling salesman problem, TSP, KP.*

I. Introduction

The knapsack problem [3] is a problem in combinatorial optimization: Given a set of items, each with a weight and a value, determine the number of each item to include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible. The problem often arises in resource allocation where there are financial constraints and is studied in fields such as combinatorics, computer science, complexity theory, cryptography [3].... . If in [8], the electronic algorithm has allowed to find an optimal solution for the travelling salesman problem, in this article, I demonstrate that the same algorithm will allow to find an optimal solution for the knapsack problem. I recall that I discovered and introduced the electronic algorithm when I was looking for an optimal solution to the travelling salesman problem :

The travelling salesman problem (TSP), which is an NP-hard problem in combinatorial optimization (see [1] , [2] , [4] and [5]), important in operations research and theoretical computer science, asks the following question: Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city? In the articles [6] and [7] , I was interested to find the Hamiltonian cycles in a graph - not necessarily optimal-. And inspired by the movement of the particles in the atom, I demonstrated (in [6] and [7]) the existence of a polynomial algorithm of the order $O(n^3)$ for finding Hamiltonian cycles in a graph. This algorithm I called Atomic algorithm (See [7]), joins other several methods to find the Hamiltonian cycles like the Monte Carlo method, Dynamic programming, or DNA computing. And to prevent memory overflow and the slow execution of the program, I suggested in [7] the use of servers: Each point x_i in the graph will be considered as a server, and each server x_i will communicate with each other server x_j with which it is connected . And finally the server x_0 will receive and display the Hamiltonian cycles if they exist.

But I found that the full of memory is caused by the fact that each server send all the list he received, hence the idea to limit the amount that will send each server, and delete as and as the list that do not serve for anything. **This idea was great, since I no longer have the problem of memory overflow and the program runs faster** : Thus only in ten minutes, with a non- powerful computer, I was able to find a Hamiltonian cycle for thousand cities. And to find an optimal solution to the TSP problem, the solution to find all Hamiltonian cycles and choose the cycle that have the shortest distance would take time hugely although servers operate simultaneously. Hence the need to seek other way to find the optimal solutions.

Thinking of the servers, we immediately think of antennas, microwave , frequencies .... in short, we think to the field of electronics. Hence the idea: If we saw servers or nodes of the graph as a node in an electrical circuit, and the "distances" between vertices as a resistor values in a circuit, then more the resistor's value is low and more the current passes quickly, so the particle or its energy will move faster by taking the shortest route.
Here, I demonstrate that the same algorithm will allow to find optimal solutions for the knapsack problem.

Definitions:

1- The 0-1 knapsack problem, which restricts the number x_i of copies of each kind of item to zero or one. Given a set of n items numbered from 1 up to n, each with a weight w_i and a value v_i, along with a maximum weight capacity W, maximize $\sum_{i=1}^{n} v_i x_i$, subject to $\sum_{i=1}^{n} w_i x_i \leq W$ and $x_i \in \{0,1\}$.
Here x_i represents the number of instances of item i to include in the knapsack. Informally, the problem is to maximize the sum of the values of the items in the knapsack so that the sum of the weights is less than or equal to the knapsack's capacity.

2- The bounded knapsack problem (BKP) removes the restriction that there is only one of each item, but restricts the number x_i of copies of each kind of item to a maximum non-negative integer value c :
maximize $\sum_{i=1}^{n} v_i x_i$, subject to $\sum_{i=1}^{n} w_i x_i \leq W$ and $0 \leq x_i \leq c$.
3- The unbounded knapsack problem (UKP) places no upper bound on the number of copies of each kind of item and can be formulated as above except for that the only restriction on x_i is that it is a non-negative integer.
Maximize $\sum_{i=1}^{n} v_i x_i$, subject to $\sum_{i=1}^{n} w_i x_i \leq W$ and $x_i \geq 0$.

An electronic algorithm to find the optimal solution for the Knapsack problem
This algorithm is almost similar to the Feynaman algorithm (See [6]) to a slight change near :
Let G be a graph on $E = \{0, \cdots, n-1\}$, where $G(i,j) = -v_j$, and the "distance" between the item i and j is noted $d_{ij} = -v_j$.
We note $\{d_1, \cdots, d_l\} = \underset{i,j}{}\{d_{ij}\}$. with $d_i \leq d_j$ if $i \leq j$.

As we have seen in the introduction, thinking of the servers, we immediately think of antennas, microwave , frequencie .... in short, we think to the field of electronics. Hence the idea: If we saw servers or nodes of the graph as a node in an electrical circuit, and the "distances" between vertices as a resistor values in a circuit, then more the resistor's value is low and more the current passes quickly, so the particle or its energy will move faster by taking the shortest route.
And there is a link between time and the "distance" :

$$t_{ij} = \frac{1}{v_{ij}} d_{ij}$$

Where v_{ij} is the speed of the electrons in the electrical circuit.

$$\omega_{ij} = v_{ij} \frac{1}{d_{ij}}$$

Where ω_{ij} is the frequency.

Otherwise seen :The server i communicate with the server j with the frequency ω_{ij} .

Time progresses, the exchange between two servers i and j in the direction $i \to j$ occurs at a frequency ω_{ij} :

To a factor, let $v_{ij} = 1$.
We shall have :

$$t_{ij} = d_{ij}$$

and

$$\omega_{ij} = \frac{1}{d_{ij}}$$

We start with the **high frequencies:** in other words by the smallest resistor value.
We first see the smallest resistor value $d_0 = d_{i,j}^0$ between i and j .

In the step s=1 (mod l):
For each i, we set $E_i^0 = w_i$.
i will give its energy to other points j through the resistors d_0.
And as in the Feynman algorithm, we construct the Feynman vector for the point j:
if $E_j = E_i^0 + w_j \leq W$

$$F(j) = \begin{pmatrix} E_j = E_i^0 + w_j \\ v(j) = -v_i - v_j \\ j \\ i \end{pmatrix}$$

Else if for one i, $E_j = E_i + w_j \geq W \ \forall j$, we add $F(j) = \begin{pmatrix} E_i \\ v(i) = -v_i \\ i \end{pmatrix}$ to H

And we construct the Feynman matrix for the point j:

$$M(F,j)=\begin{pmatrix} E_j & \cdots \\ v(j)=-v_i-v_j & \cdots \\ j & \cdots \\ i & \cdots \end{pmatrix}$$

So,like in [7], we will have :

$$M(F,j)=\begin{pmatrix} E_j & \cdots \\ v(j) & \cdots \\ \vdots & \cdots \end{pmatrix}$$

In the step s=2 (mod l):

Among the points i related to j we choose the minimum resistor value d_{s1} such that $d_1 > d_0$ if it exist, else we choose the minimum resistor value d_1 such that $d_1 \leq d_0$

Each point i linked to a resistor with value d_1 will give its energy to other points through the resistor d_1.

In the step s+1 (mod l):

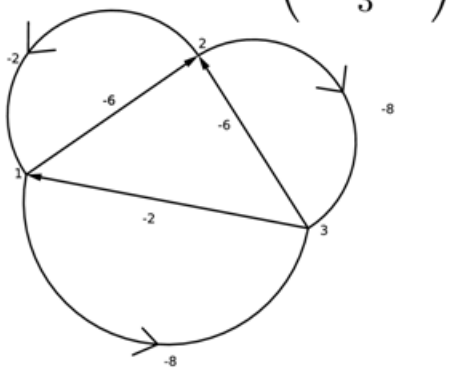
We continue this process and as the energy increases and must be less than W; H will not be empty, and the algorithm, as the Feynman algorithm, will stop, and will display the first vector in H with a shorter "distance"(the optimal solution for the Knapsack problem).

Example:

We can check this technique for example for the Graph G on the basis $E=\{1,2,3\}$ with : $w_1=1$,$w_2=5$, and $w_3=3$ and with the values (or "distances") $v_1=d(1,1)=2$, $v_2=d(2,2)=6$, $v_3=d(3,3)=8$ and $W\leq 8$.

We check that the vector ; $F(2)=\begin{pmatrix} E_2=8 \\ -14 \\ 2 \\ 3 \end{pmatrix}$ is the first Feynman vector that enters in H. which gives the optimal solution $\{2,3\}$ with 14 for value.

$$\mathcal{F}(2)=\begin{pmatrix} E_2=8 \\ -14 \\ 2 \\ 3 \end{pmatrix}$$



Example of the Knapsack problem

Note:

1- The electronic algorithm [8] was used both for finding the optimal Hamiltonian cycles if they exist, and for finding the optimal solution for the Knapsack problem(See the example above).

2- By making restrictions while the construction of the Feynman vectors, this algorithm solves simultaneously the three versions of knapsack problem above : The 0-1 knapsack problem,The bounded knapsack problem (BKP) and the unbounded knapsack problem (UKP).

References

[1]. Lizhi Du. A polynomial time algorithm for hamilton cycle. IMECS, I: 17–19, March 2010. Hong Kong.
[2]. L.Lovasz. Combinatorial problems and exercises. Noth-Holland, Amsterdam, 1979.
[3]. R.C. Merkle and M.E. Hellman. Hiding information and receipts in trap door knapsacks. Internal Symposium on Information Theory, université Cornell, Ithaca, New York, October 1977.
[4]. D.S.Johnson M.R.Garey. Computers and intractability:a guid to the theory of np-completeness. Freeman, San Francisco, 1979.
[5]. R.Diestel. Graph theory. Springer, New York, 2000.
[6]. M. Sghiar. Algorithmes quantiques, cycles hamiltoniens et la k-coloration des graphes. Pioneer Journal of Mathematics and Mathematical Sciences, 17-Issue 1:51–69, May 2016.
[7]. M. Sghiar. Atomic algorithm and the servers' s use to find the hamilto-nian cycles. International Journal of Engineering Research and Applications (IJERA), ISSN: 2248-9622, 6-Issue 6:23–30, jun 2016.
[8]. M. Sghiar. An electronic algorithm to find the optimal solution for the travelling salesman problem. IOSR Journal of Mathematics (IOSR-JM),e-ISSN: 2278-5728, p-ISSN: 2319-765X, 12:82–86, August 2016.