BIN PACKING
Proof of NP Completeness and Hardness.

Bhuvaneshwaran Ravi , Kameswaran Rangasamy, Serlin Tamilselvam.

Bin Packing:

Having an infinite number of bins each of volume V, can we put a set of n objects of volumes vi , 1 ≤ i ≤ n into less than k bins for a given constant k?

Bin Packing belongs to NP and thus NP-Complete:

In order to consider a problem to be NP, it should be able to verify the problem in polynomial time*[1]*. In our case let us assume,

A - be the size certification.

$S = \{s_1, s_2, s_3 .... s_n\}$ be the set of objects to be put into bins each of volume $\mathbf{v_i}$.
T be the total volume of all the object.

k – number of bins used (or) number of subsets the objects were devided.
$\{k_1, k_2, k_3, k_{4 ....} k_n\}$ be the bins.

The problem is to devide and put all the set of objects into subsets and optimally put the subets into bins.

So,

summation of the volume of all the objects = summation of volume of k bins used = T *[3]*.

That is,

$\mathbf{\Sigma_{i=A} k_i = \Sigma s_i = T}$

This gives us a polynomial verfication certification.

And thus the Bin Packing problem is NP-complete.

NP-Hardness of Bin Packing :

In order to proove Bin Packing is NP-Hard we are reducing it from another NP-hard problem *[1]* i.e Partion Problem.

Approximation algorithm and NP-Hardness proof*[2]*:

Given an instance i1,i2,i3......in of the PARTITION problem let us consider that,

$I = \boldsymbol{\Sigma i_j}$

and consider an input instance of the Bin Packing to be $s_j = 2i_j / I$.

The above equation says that ,

On a given instance of BIN PACKING split the instances into two that summation of both the sub-sets are equal, given that the element from one of the sub-set does not occur in the other.*[2]*

This instance prooves that the answer to the PARTITION problem is TRUE if and only if the number of bins to pack of BIN PACKING problem is 2.*[2]*

If in case there exists a k-approximation algorithm which would solve Bin Packing on polynomial time ($k<3/2$) then the answer to the above problem "2 bins" would be always found.*[2]*

If the algorithms even takes 1 more than 2, i.e 3 bins then the optimal factor would be greater than 3/2. *[2]*

Hence, the algorithm would solve the PARTITION problem in polynomial time which directly say than P=NP.

BIN PACKING can then be solved in polynomial time. Unless P=NP *[1]*.

Therefore, BIN PACKING is an NP-Hard problem with approximation value > 3/2 .*[2]*

Partition problem :

Given an instance a1, a2, a3,..... an of non-negative integers, decide whether there is a sub-set X such that,

$\boldsymbol{\Sigma}_{\mathbf{i=S}}\ \mathbf{a}_i = \boldsymbol{\Sigma}_{j=S}\ \mathbf{a}_j$

Split the integers into two subsets where an item from one of the subset does not belong in the other. The sum of one of the subset will be equal to the sum of the other one.

The PARTITION problem has already proven to be NP-Hard.

REFERENCES:

1. Introduction to Algorithms by Thomas H. Cormen, Charlese E. Leiserson.
2. Bin Packing by Ecole Polytechnique Federale de Laussane
3. Bin Packing by Dr. Nina Amenta from University of California.

/