Study of Bin packing algorithm

Nitish Rawat
Guided by Dr. Priya Chandran

National Institute of Technology, Calicut

December 6, 2013

Contents

- Introduction
- Motivation
- Project aim
- Literature survey
- Work done
- Conclusion
- Future work

Introduction

Task scheduling uses combinatorial algorithms. We try to optimize task scheduling in cluster. Heterogenous computer cluster contains computers of different functional units. For eg: laptop, tablet, game console, smart television, mobiles, and similar computing devices etc.
Bin packing is mathematical model of task-to-processor assignment. This is a NP-hard problem. We try to optimize this problem using polynomial linear program. We also find why it is infeasible?

Motivation

Resource scheduling in computer cluster is known as resource pooling. Clusters are collection of computers or computing devices. Sharing resources for tasks/processes in cluster are needed. Optimum utilization if resources are the primary motivation.

Problem Statement

Bin Packing Problem [1]

Bins are containers to store items. Given n items with sizes $a_1, \ldots, a_n \in (0,1]$, find a packing in unit-sized bins BIN_1, BIN_2, ... that minimizes the number of bins used.

Project Aim

The aim of the project is to implement algorithm for Bin packing.

Literature Survey

There are many task-to-processor algorithms found in literature. Bin packing algorithm has been a very old problem. Naive algorithm exist from ancient times. Linear programming was formed in 1947. There was an attempt to solve bin packing using linear program in 2012. ▸ Limitations of on-line algorithms

Algorithms found during Literature Survey

- ▸ Next-Fit Algorithm
- ▸ First-Fit Algorithm
- ▸ First-fit decreasing Algorithm
- ▸ Linear Program formulation
- ▸ Algorithm of interest

Algorithms so far

Next-Fit Algorithm [2]

Step 1. Check the open bins if the item can be placed. If yes, place it and proceed to the next item.

Step 2. Else, close the oldest bin, open a new one and go back to step 1.

back

Algorithms so far

First-Fit Algorithm[2]

Step 1. Check the open bins to pack the item. If placed then move to next item.

Step 2. If item does not fit into any of these bins, it opens a new bin.

back

Algorithms so far

First-Fit Decreasing Algorithm[2]

Step 1. Sort the items in decreasing order.

Step 2. Check the open bins to pack the item from begining. If placed then move to next item.

Step 3. If item does not fit into any of these bins, it opens a new bin.

back

Formulation of bin packing[3]

Our objective is to minimize the number of bins used. We formulate the integer program as follows -

$$\text{minimize} : \sum_{j=1}^{m} y_j \tag{1}$$

such that

$$\sum_{j=1}^{m} x_{ij} = 1 \qquad 1 \leq i \leq n \tag{2}$$

$$\sum_{i=1}^{n} p_i . x_{ij} \leq 1 \quad 1 \leq j \leq m \tag{3}$$

$$y_i \geq x_{ij} \qquad 1 \leq i \leq n, 1 \leq j \leq m \tag{4}$$

$$x_{ij} \in \{0, 1\} \quad 1 \leq i \leq n, 1 \leq j \leq m \tag{5}$$

Formulation of bin packing[3]

The constraints of the problem are as follows -

- Constraint (2) states that every vector is packed in a bin.
- Constraint (3) ensures that the packed vectors do not exceed the bin capacity.
- Constraint (4) tells whether a bin is used or not.
- Constraint (5) ensures that a vector is either packed entirely in a bin or not.

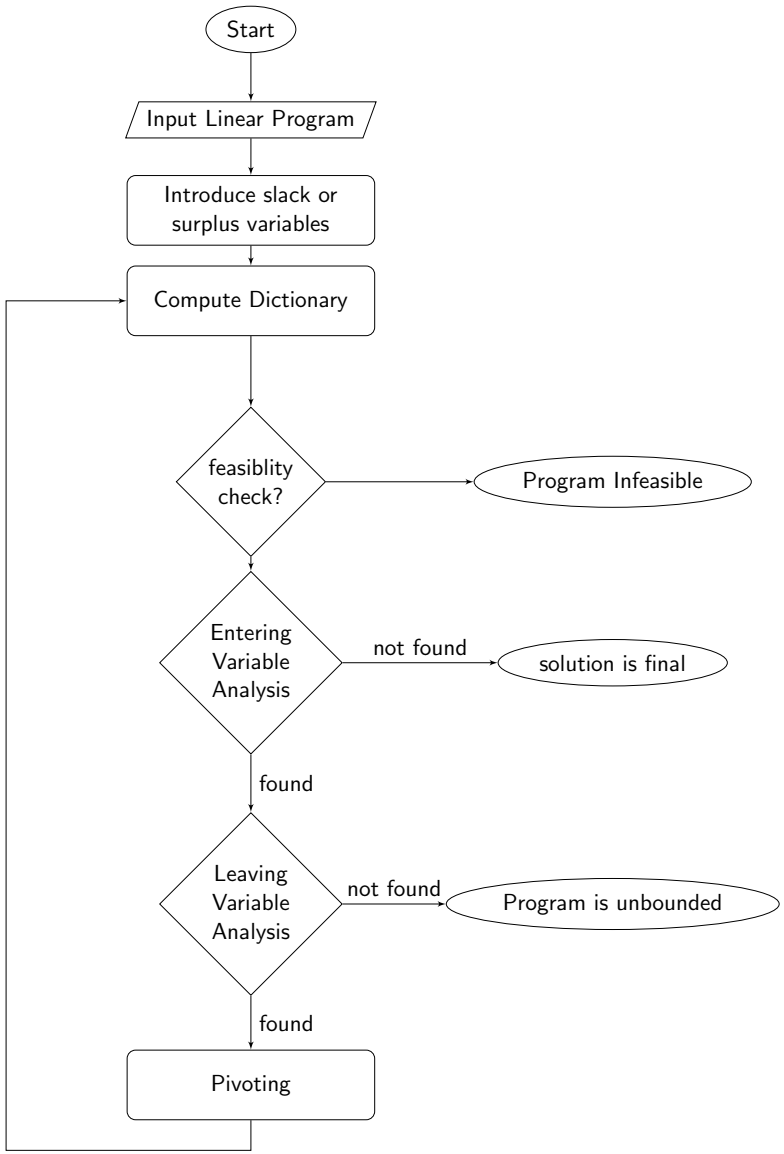
back

Drawbacks/limitations of previous algorithms

There exist inputs that can force ANY online bin-packing algorithm to use at least 4/3 times the optimal number of bins.

back

Design of Simplex Method[4]



Algorithm for entering variable analysis[4]

Data: Linear program dictionary
Result: Entering variable index
//C is objective function coefficient;
while $(C_i) \in C$ **do**
    if $(C_i) > 0$ **then**
        Entering_Variable_Index = i;
        break;
    end
end

Algorithm 1: Entering Variable Analysis

Algorithm for leaving variable analysis[4]

Data: Linear program dictionary
Result: Leaving variable index
//b is RHS of basic variables;
//A is coefficient of constraints;
for $i \in b_i$ **do**
  if $A_{i,entering\_variable} < 0$ **then**
    $bound_i = b_i/(-A_{i,entering\_variable})$
  end
end
Leaving_Variable_Index = which.min($bound$);

Algorithm 2: Leaving Variable Analysis

Pivoting[4]

Data: Linear program dictionary (D')
Result: Linear program dictionary (D'')
$//D', D''$ is dictionary of linear program;
$pivotElement = A_{leaveVariable,enteringVariable}$;
// Invert the row of pivotElement;
$A_{leaveVariable,} = A_{leaveVariable,}/(-pivotElement)$;
$A_{leaveVariable,enteringVariable} = 1/(pivotElement)$;
$b_{leaveVariable} = b_{leaveVariable}/(-pivotElement)$;
//replace the leaving Variable in constraints;
for $b_i \in b \wedge i \neq leaveVariable$ **do**
    $temp = A_{i,enteringVariable}$;
    $A_{j,} = A_{j,} + temp * A_{leaving_Variable,}$;
    $A_{j,entering_Variable} = temp/pivotElement$;
    $b_j = b_j + temp * b_{leaving_Variable}$;
end

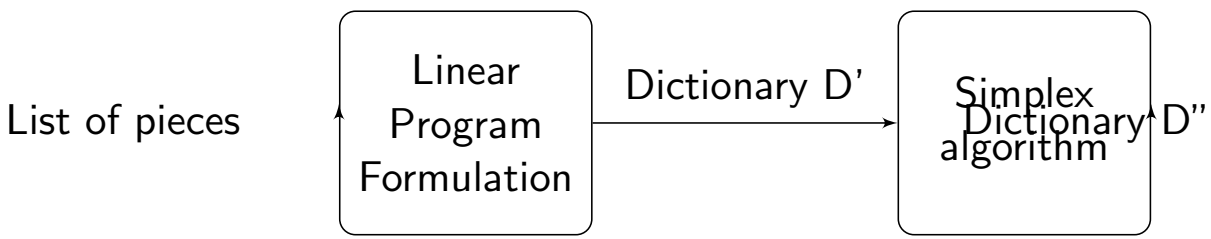
Algorithm 3: Pivoting

Implementation

A program is written in R. The program was used to solve linear programming problem using Simplex Method.

Block diagram

- Task scheduling is a computational process.
- Task scheduling is reduced to a mathematical problem called bin packiing.
- Bin packing is formulated in linear program
- linear program is solved using simplex algorithm.



Dictionary[4]

Dictionary (D) of Linear Program for the list (L). The dictionary is a data-structure for linear program. Dictionary is a representation of solution of linear program. Dictionary (D) contains the following data:

- m = number of basic variables
- n = number of non-basic variables
- B = indices of basic variables (basis set)
- N = indices of non-basic variables (independent set)
- b = RHS of constraints
- A = matrix of coefficients of rules(constraints)
- z0 = solution of the linear program
- C = co-efficients of objective function

Dictionary of Bin Packing[4]

10 6 # m n
7 8 9 10 11 12 13 14 15 16 #B
1 2 3 4 5 6 #N
1 1 -1 -1 1 1 0 0 0 0 #b
0 0 -1 -1 0 0 #A
0 0 0 0 -1 -1
0 0 1 1 0 0
0 0 0 0 1 1
0 0 0.1 0 0.8 0
0 0 0 0.1 0 0.8
1 0 -1 0 0 0
1 0 0 -1 0 0
0 1 0 0 -1 0
0 1 0 0 0 -1
0 -1 -1 0 0 0 0 # z0 C

Input Set 1

First set of input is following:
<0.1>
<0.8>

Input Set 2[6]

First set of input is following:
Another example[5] is following:
<0.2>
<0.5>
<0.4>
<0.7>
<0.1>
<0.3>
<0.8>

Input Set 3

First set of input is following:
Another example is following:
<0.1>
<0.5>
<0.1>

Research Conclusion

Why it fails?

The linear program of bin packing is infeasible because the R.H.S. of the basic variable do not satisfies the constraintS that all the variables should be greater than or equal to 0. But when we form the dictionary of bin packing there are negative values in R.H.S. **NP**-hard optimization problemss was tried using **P** linear program. The program was found to be infeasible. This is concluded.

Future Work and Scope

The linear problem may be converted to non-linear program. A **software for mixed constraint non-linear programming problem** can be written. A simulation of input may be tried on software. This simulation may attempted to run on cluster.
The algorithm may be **further optimised** if possible.
The lower bound of off-line bin packing problem is not known. A future work can be done for finding a lower bound for bin packing problem.
This problem may solve the instance of **task scheduling on cluster**. Heterogenous computer are machines which have different available resources. A heterogenous cluster may be scheduled.
A **machine (finite state or infinite state)** may be designed for solving NLP of task scheduling. Such machine may be concatinated with heterogenous cluster or LAN to utilize full computing resources of network.

References I


W. Fernandez de la Vega and G. Lueker.
Bin packing can be solved within 1 + epsilon in linear time.
Combinatorica, 1:349–355, 1981.
10.1007/BF02579456.


Vazirani and Vijay V.
Approximation algorithms.
Springer-Verlag New York, Inc., New York, NY, USA, 2001.


Chandra Chekuri and Sanjeev Khanna.
On multi-dimensional packing problems.
In *Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '99, pages 185–194, Philadelphia, PA, USA, 1999. Society for Industrial and Applied Mathematics.

References II


Shalom D. Ruben Sriram Sankaranarayanan.
Linear and integar programming.
2013.


Subhash Suri.
Data structure and algorithms.
2012.


Andrew Chi-Chih Yao.
New algorithms for bin packing.
J. ACM, 27(2):207–227, April 1980.


Marguerite Frank and Philip Wolfe.
An algorithm for quadratic programming.
Naval Research Logistics Quarterly, 3(1-2):95–110, 1956.

References III


Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein.
Introduction to Algorithms.
2011.


Chetan Rao.
Technical report, University of Wisconsin-Madison, 2012.

THANK YOU