THOMSON REUTERS ENDNOTE

[Bae* *et al.*, 5(10): October, 2016]
IC™ Value: 3.00

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

IJESRT
INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

ALGORITHM OF PACKING ITEMS INTO PATH-GRAPH WITH CONSTRAINTS ON ADJACENT NODES

Jungryul Bae[1], Junhyuk Park[2], Hyung Tag Han[3], Ho Rim Lee[3], Yongju Cho[4], Hyunchul Tae[4*]
[1] Department of Mathematics, Yonsei University, Republic of Korea
[2] WM Logistics, United States of America
[3] Department of Mechanical Engineering, Konkuk University, Republic of Korea
[4] Korea Institute of Industrial Technology (KITECH), Republic of Korea

DOI: 10.5281/zenodo.160883

ABSTRACT

A path-graph is a graph that can be arranged in a linear sequence that two nodes are adjacent if they are consecutive in a sequence, nonadjacent otherwise [1]. In this paper, we will think of the situation that there is a path-graph of p nodes and a set of k positive numbers which are less than a constant number d. Every node in the graph will be packed with a number in the set satisfying following constraints. First, sum of numbers in two adjacent nodes must be less than d. And first and last node has number limits that will be packed in each nodes. Then we want to check if the set of numbers is feasible to the path-graph or not. We will call the k positive numbers 'items'. The classic method of checking all possible cases can be done. But it takes too much computing time in the worst case. Algorithm we introduce reduces the complexity from $O(p!)$ to $O(p^2)$.

KEYWORDS: Graph Theory, Path-Graph, Packing Graph, SMT, SMD, Gantry

INTRODUCTION

A path-graph is a graph that can be arranged in a linear sequence that two nodes are adjacent if they are consecutive in a sequence, nonadjacent otherwise [1]. In this paper, we will think of the situation that there is a path-graph of p nodes and a set of k positive numbers which are less than a constant number d. Every node in the graph will be packed with a number in the set satisfying following constraints. First, sum of numbers in two adjacent nodes must be less than d. And first and last node has number limits that will be packed in each nodes.

We will call the k positive numbers 'items' from now on. The problem is "can all items in the set be packed in the graph?" In other words, we want to check if the set of numbers is feasible to the path-graph or not. When$k >$ p, it is obviously infeasible. So checking the case of $k \leq p$ is enough. The classic method of checking all possible cases can be done. But it takes too much computing time and our algorithm successfully reduces the time.

This problem was discussed during our ongoing study about routing SMD gantry. SMD is a device that assembles electric components on PCB (printed circuit board). We studied about piano-type multi-headed gantry which is a gantry that has consecutive heads in a row with same intervals [2]. The components that should be mounted on gantry has different size and this may cause a collision between consecutive components. This was calculated by classic inefficient methods in real world industries.

We found out that this problem can be thought as the problem of graph theory and defined as above. Since this problem is quite simple and has real world application, we concluded that it has some significant meanings and decided to introduce it.

In this paper, we will first introduce some similar problems. And we will explain the algorithm and its validity. Lastly, we will end with our conclusion of the study.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

LITERATURE REVIEW

We found some problems which is related with our study.

Four color theorem is a theorem which states that four colors are enough to color every planar map. Four color problem is equivalent with the problem that every adjacent nodes of every planar graph without loops can be distinguished with each other for maximum four colors [3]. This was proved by K. Appel and W. Hanken [4] by distinguishing every cases and using computer.

Ménage problem is about finding the number of ways of seating n man-woman couples to the circular table with two constraints that is first, men and woman alternates, and second, partners doesn't sit next to each other [5]. This problem can be said as sequencing people into cycle graph. Many solutions of this problem has been proposed.

These problems and our problem are about packing elements into the graph with the properties that an element that is packed in a node limits the elements on adjacent nodes. Especially the Ménage problem deals with cycle graphs, which has some characteristics similar to path graphs. But our problem allows empty nodes and different number of nodes and elements. Also criteria of limiting adjacent nodes are different. So our algorithm won't be an extension of previous studies and will be a separate study.

ALGORITHM

Goal of the algorithm we suggest is finding the best packing method to make an item set feasible to the path graph with constraints. Thus, if its infeasible after the algorithm, there is no better way to pack that item set into the graph.

1. Assumptions

- One node can be packed with maximum of one item only.
- Size sum of two items that is packed on two adjacent nodes should be under a constant ($\leq d$).
- Size of items are under d.
- There is extra constraint on the item sizes of the leftmost and rightmost nodes.

We will describe this problem in other words for readers to understand more easily.

2. Equivalent description of the problem

- We can assume that path graph is placed on the straight line of left-right direction and distance between eve ry adjacent nodes is d.
- We will say that item with size r as a circle with radius r and the center of the items sticks to the node whe n it is packed.

Then the problem is transformed into packing set of circle items into nodes at the center such that any two items never overlap.

Also, to make the algorithm clear, we will define a specific item type which is called zero item. Zero items are an imaginary items that has size of zero but can occupy one node. If a zero item is packed on a node, it means that the node will be left empty.

3. Notations

- p : # of nodes
- $n_1,\ n_2,\ \cdots,\ n_p$: nodes (from the left)
- b_L, b_R : size limit of leftmost (n_1) and rightmost (n_p) nodes
- d : distance between two adjacent nodes
- k : # of items
- $T_i\ (i = 1,\ 2,\ \cdots,\ k)$: items in ascending order of radius (same notation for their radius ; $0 \leq T_1 \leq T_2 \leq \cdots \leq$

$T_k < d$)

- $I = \{T_1, \cdots, T_k\}$: Set of all items which will be packed
- Z_i $(i = 1,\ 2,\ \cdots,\ p)$: zero items (same notation for their radius ; $0 = Z_1 = \cdots = Z_p$)
- $I_0 = \{Z_1, \cdots, Z_p\}$: Set of p imaginary zero items

4. Algorithm

Simply saying, this algorithm is iteration of 'packing the rightmost empty node with largest packable item in $I \cup I_0$'.

- Pseudo code

$function\ select\_item(items\ set\ Y, length\ b)$

$return\ 'the\ largest\ item\ T\ in\ Y\ such\ that\ T < b'\ as\ the\ output\ of\ the\ function$

main()

{

$for(j = p; j \geq 1;\ j - -)$

$if(j \geq 2)$

$item\ C =\ select\_item(I \cup I_0, b_R)$

$mount\ \ C\ on\ n_j$

$b_R = d -\ C$

$if(j = 1)$

$mount\ select\_item(I \cup I_0, \min(b_L, b_R))on\ n_j$

}

If this algorithm is done with a set of items I and p nodes, we say that I is best-packed onto p nodes. This algorithm has time complexity of $O(p^2)$.

5. Feasibility

Let I' is a set of remaining items of I after the algorithm. Then if I' is empty after the process, the original item set is feasible. Otherwise, infeasible. We will speak more precisely.

- Sub-statement : For all $k \in N$ (natural number), assume that $I = \{T_1, \cdots, T_k\}$ (set of any k items) is best-packed into p nodes with boundary conditions b_L and b_R. Then, I is infeasible if and only if I' (set of remaining items of I after it is best-packed) is not empty.
- Statement : For all $p \in N$, sub-statement is true.

The statement can be proved by mathematical induction. It is enough to prove that I is infeasible for p nodes if I' is not empty. The statement in other direction is obviously true.

- Proof of the statement

i) When $p = 1$, Sub-statement is obviously true.

ii) When $p = P$, Assume that sub-statement is true.

iii) When $p = P + 1$, Let I' is not empty.
If $P + 1$ nodes $n_1,\ \cdots,\ n_P, n_{P+1}$ are best-packed, left P nodes $n_1,\ \cdots,\ n_P$ are best-packed also. It mea

THOMSON REUTERS ENDNOTE

ISSN: 2277-9655
[Bae* *et al.*, 5(10): October, 2016] Impact Factor: 4.116
IC™ Value: 3.00 CODEN: IJESS7

ns left P nodes are infeasible by assumption on ii).

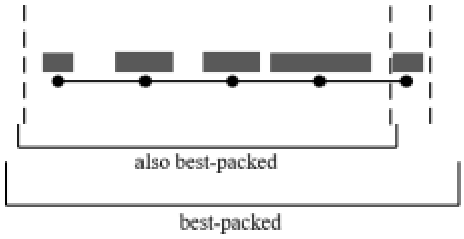

Figure 1. Inductiveness of the algorithm

a. If zero item is packed on n_{P+1} (because other nonzero items were not packable to n_{P+1}), left P nodes are best-packed with I. By assumption of ii), left P nodes are infeasible. To make them feasible, other item set which is different from I should be packed on left P nodes. This means that an item of I should be packed on n_{P+1} and its impossible.

b. If nonzero item T_j is packed on n_{P+1} (because $T_{j+1}, \cdots, T_k$ are not packable to n_{P+1}), left P nodes are best-packed with $I/\{T_j\}$. By assumption of ii), left P nodes are infeasible. To make them feasible, other item set which is different from $I/\{T_j\}$ should be packed on left P nodes. This means that an item of $I/\{T_j\}$ should be packed on n_{P+1}. So we pack T_i (or a zero item) which satisfies $T_i < T_j$ on n_{P+1} and assume that there is a feasible packing sequence in this condition. We will notate the node that holds T_j as n_t in this feasible sequence. Then if we reverse the sequence of items from n_t to n_{P+1}, n_{P+1} holds T_j as it was at the first time and it is also feasible. It makes contradiction since assumption of ii) makes the reversed sequence infeasible.

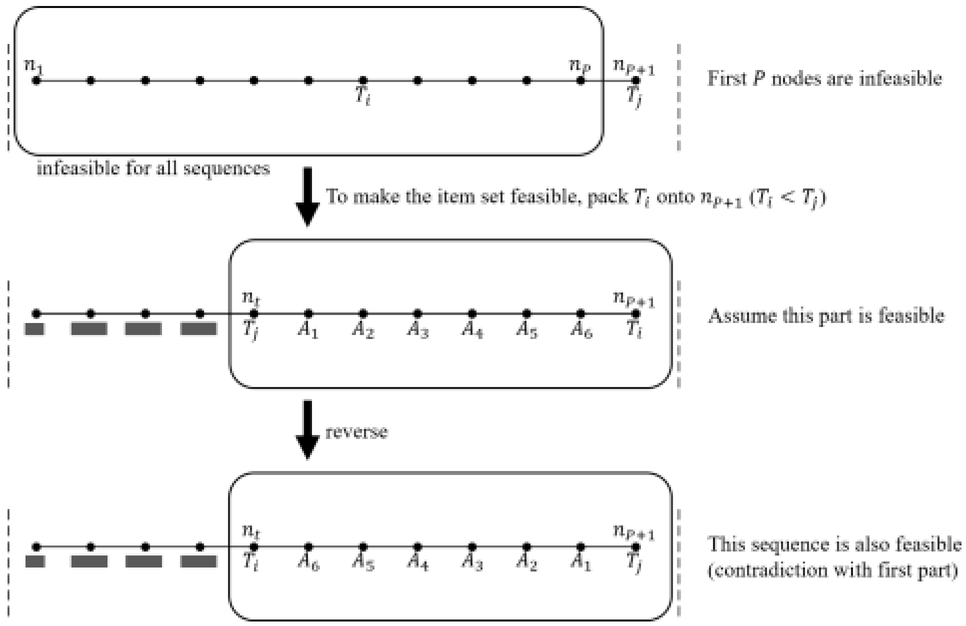

Figure 2. Diagram of iii)-b.

This shows that if we assume that sub-statement is true when $p = P$, sub-statement is also true when $p = P + 1$.

Three steps conclude that the statement is true. That is, following the algorithm is actually the best way to make the set of items feasible to the graph.

DISCUSSION

This was the case of path graph and we can also think of the cycle graph. If we pack any item at any point of the cycle at the first time, the problem becomes just as the case of path graph with left and right boundary conditions. So we can execute the same algorithm to check the feasibility of the cycle graph.

CONCLUSION

This paper defines the item packing into path-graph problem and introduces the exact algorithm to check the feasibility of the item set. It reduces the complexity from $O(p!)$ to $O(p^2)$. One of the applications can be SMT gantry mount feasibility problem and we will study further on this topic.

ACKNOWLEDGEMENTS

This study has been funded by following two projects of KITECH (Korea Institute of Industrial Technology).

- Simulation Based Sewing Process Optimization Algorithm
- A Development of the KLM system for the Stabilization of Kimchi Raw material and study on Food Industry

REFERENCES

[1] J. A. Bondy and U. S. R. Murty, *Graph theory with applications*, vol. 290. Citeseer, 1976.

[2] H. Tae and B.-I. Kim, "Feeder Re-assign Problem in a Surface Mount Device with a Piano-Type Multi-Headed Gantry," *Ind. Eng. Manag. Syst.*, vol. 12, no. 4, pp. 330–335, 2013.

[3] K. Appel and W. Haken, "Every planar map is four colorable," *Bull. Am. Math. Soc.*, vol. 82, no. 5, pp. 711–712, 1976.

[4] K. Appel, W. Haken, and others, "Every planar map is four colorable. Part I: Discharging," *Illinois J. Math.*, vol. 21, no. 3, pp. 429–490, 1977.

[5] K. P. Bogart and P. G. Doyle, "Non-sexist solution of the m{é}nage problem," *Am. Math. Mon.*, vol. 93, no. 7, pp. 514–518, 1986.