UC San Diego
Technical Reports

Title

NP-Completeness and Approximation Scheme of Zero-Skew Clock Tree Problem

Permalink
https://escholarship.org/uc/item/9wg693jj

Author
Liu, Bao

Publication Date
2005-10-13

Peer reviewed

eScholarship.org Powered by the California Digital Library
University of California

NP-Completeness and Approximation Scheme of Zero-Skew Clock Tree Problem

Abstract: Routing Zero-Skew Clock Tree with minimum cost is formulized as Path-length Balanced Tree (PBT) Problem. Various kinds of heuristics have been proposed. But people don't know the exact nature of the problem. We prove PBT problem is NP-Complete in Manhattan, Euclidean and diagonal plane, and give an approximation scheme for path-length delay model with $N^{O(1+c\lg N)}$ time to achieve (1+1/c) OPTIMUM.

1. Introduction

Most VLSIs of today use a global clock signal to syncronize the circuit. This clock signal is usually embedded as a tree structure. There are two major concerns of this tree structure: one is it should take equal time for the clock signal to propogate from the source to any of the sinks, since the clock's purpose is to syncronize. We call this path-length balance condition. Noticing that under path-length delay model the delay is propotional to the geometrical path length between the source and the sink. While under Elmore delay or higher order delay model this path-length has other definitions[1].

The other concern is that this clock signal consume a huge part of the total power of the circuit, due to its nature of being the most active part of the circuit, and its large capacitance[2].

So the problem comes: how to construct a clock tree with minimum capacitance while satisfying the path-length balance condition. This problem has been well studied and various heuristics have been proposed, among them famous are the H-tree[3], MMM[4], Top-down Planar Partition[5], Bottom-up Geometric Max-Matching[6] and Clustering[7]. While the exact nature of the problem remained open[8][6]. People tend to believe it's NP-hard.

In this paper we prove PBT problem is NP-Complete in Manhattan, Euclidean and diagonal plane(where the distance between two points $d(v_1,v_2)=|x_1-x_2|+|y_1+y_2|$, $d(v_1,v_2)=((x_1-x_2)^2+(y_1-y_2)^2)^{1/2}$ and $d(v_1,v_2)=|x_1-x_2|+|y_1-y_2|+(\sqrt{2}-2)\mathrm{Min}(|x_1-x_2|,|y_1-y_2|)$, respectively). Further we give an approximation scheme to achieve (1+1/c)OPTIMUM in $N^{O(1+c\lg N)}$ time for PBT problem with path-length delay model, using Sanjeev Arora's method[11][12]. The paper is organized as: section 2 gives the problem formulation, section 3 gives the NP-Completeness proof, section 4 gives the approximation scheme, and section 5 gives the conclusion.

2. Problem Formulation

Definition 1:
PBT problem:
Instance: *a set S of nodes in a metric space , positive rational number B.*
Qestion: *Is there a set P of points labeled with levels such that the embedded PBT has cost ≤ B?*
The PBT is constructed by connecting the nodes in set S to the closest nodes with deep most level in set P, and connecting the nodes with level i to the closest nodes with level i-1 in set P. The cost of PBT is the summary of the costs of all connections, under path-length balance condition.
To give a mathematical formulation,

$$\sum_{p \in Pk} \overset{\Sigma MAX}{_{v \in S \cap C_p}} \overset{MIN}{_{u \in P_k}} d(u,v) + \sum_{i=1}^{k-1} \sum_{p \in Pi} \overset{\Sigma MAX}{_{v \in P_{i+1} \cap C_p}} \overset{MIN}{_{u \in Pi}} d(u,v) \le B$$

where $C_p = \left\{ u \middle| \overset{MIN}{_{v \in P_i}} d(u,v) = d(u,p) \right\}$

and P_i is a subset of P of points with level i.

3. NP-Completeness Proof

Definition 2:
PBT_h (PBT with < h levels) **problem:**
Instance: *a set S of nodes in a metric space with distance metric d(), integer h, positive rational number B.*
Qestion: *Is there a set P of points in the metric space with levels ≤ h such that the embedded PBT has cost ≤B?*

Lemma 1: $PBT_h \leq_p PBT_{h+1}$.

Here the symbol $\leq_p$ denotes polynomial-time reduction. The lemma says given instance G, an instance G' can be constructed in polynomial time such that instance G has a polynomial-time solution of PBT_h with cost B if and only if instance G' has a polynomial-time solution of PBT_{h+1} with cost B'. Hence if a polynomial-time algorithm for PBT_{h+1} problem exists, so does a polynomial-time algorithm for PBT_h problem .

Proof:

Construct:

G' can be constructed by having a duplicated graph G_d in a distance of L away from the original graph G, where L is large enough such that the distance between any two points in G and G_d, is larger than the distance between any two points within G or G_d. We constructe a PBT_{h+1} by constructing PBT_h in G and G_d, then connecting the root of each PBT_h together, and locate a new root in the middle (Fig.1).

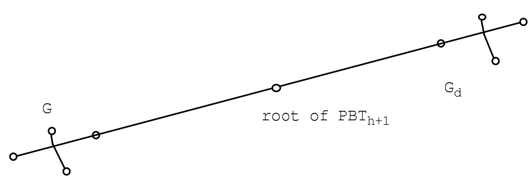

Fig.1 construct a duplicated graph G_d and PBT_{h+1}

Optimality:

Given L is large enough, since any partition which has some nodes in G and some nodes in G_d will has arbitrary large diameter, it's guaranteed suboptimal. So the only optimal partition on the first level is to partition into subsets of G and G_d. We can not reduce the cost by changing the partition on the first level.

We also cannot reduce the cost by moving the root of G or G_d. Moving the root of G by a distance of δ will increase the cost by $(N_G - 2)\delta$, where N_G is the number of nodes in G, since moving will increase the distance from each node of G to the root of G by δ, and decrease the distance from the new root to both the root of G and Gd by δ. Assuming $N_G > 2$, moving can only make the cost increased. Cases of $N_G \leq 2$ are trivial.

So the partition on the first level is optimal and the location of the roots on the first level are optimal. The optimality of the two PBT_h's guarantee the structures underneath are optimal. So we conclude the PBT_{h+1} so constructed is optimal.

Polynomial Reduction Correctness:

Since G and G_d are identical, their PBT_h's are also identical, and the distance between the roots of each PBT_h is exactly L. Setting B' = B+L, we have G has a PBT_h of cost B if and only if G' has a PBT_{h+1} of cost B'.□

PBT_1 is trivial to solve by locating the root in the middle of the longest path between nodes. Next we show PBT_2 is NP-Complete so for all i>2 PBT_i is also NP-Complete.

Definition 3:

PBT_2:

Instance: *a set S of nodes in a metric space with diameter R, positive rational number B.*

Qestion: *Is there a set P of K subsets C_i, such that if r_i is the diameter of partition i, n_i is the number of nodes in partition i, then $\Sigma_i (n_i-1)r_i + KR \leq B$?*

To prove PBT_2 is NP-Complete, we first need the NP-Completeness proofs of the following problems, which are similar but different with the p-center and p-median problems[9].

Definition 3:

N_i-1 Weighted-Sum-Diameters K-way Partition Problem:

Instance: *a set S of nodes in a metric space, positive integer k, positive rational number B.*

Qestion: *Is there a k-way partition of set S, such that if C_i is a subset, d_i = Max d(u, v)| u&v $\in$ C_i is the diameter of subset C_i, n_i = $|C_i|$ is the number of nodes in subset C_i, then $\Sigma_i (n_i-1) d_i \leq B$?*

Lemma 2: N_i-1 WSD K-way Partition Problem in Manhattan plane is NP-Complete.

Proof:

We prove this by polynomial-time reduction from 3-Satisfiability [10]. That is, we construct a set of nodes in Manhattan plane, and solve the 3-SAT problem by solving the N_i-1 WSDKP Problem. Since 3-SAT is NP-Complete, so is N_i-1 WSDKP Problem.

Definition 5:

3-SAT:

Instance: *a set U of variables, collection C of clauses over U, each with exactly 3 literals.*

Qestion: *Is there an satisfying assignment of U for C ?*

Construct & Optimality:

We construct a circuit of nodes representing the truth assignment of each variable(Fig.2). The number of nodes in a circuit is noted as S_i. S_i mod 3 = 0. For nodes in the circuit, if v_i mod 3 = 1, we have $d(v_i, v_{i+1}) = 1$, otherwise we have $d(v_i, v_{i+1}) = b >> 1$. Denoting d(i) as the minimum diameter of i nodes in such a circuit, we have

d(1) = 0, d(2) = 1, d(3) = b+1,

d(4) = 2b+1, d(5) = 2b+2, d(6) = 3b+2 ...

So to partition a circuit of S_i nodes into $S_i/3$ subset, such that $\Sigma_i\ (n_i\text{-}1)d_i$ is minimum, there are only two optimum solutions. One is to cluster $\{v_i, v_{i+1}, v_{i+2}\}$ to form a subset for each i mod 3 = 1, the other is to cluster $\{v_i, v_{i+1}, v_{i-1}\}$ to form a subset for each i mod 3 = 1. These two partitions correspond to the truth assignment of the corresponding variable in 3-SAT problem.

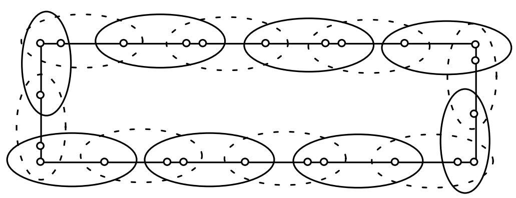

Fig.2 a circuit of nodes in Manhattan plane
(the two partitions represent the true/false assignment of the correponding variable)

We construct for each clause a "clause structure", which is formed by a "clause node", and nearby nodes of three circuits (Fig.3). The distance between a clause node and the nearest circuit node is b. Our objective is to partition the whole set of nodes in the plane into $\Sigma\ S_i/3$ subsets. Denoting m(i) as the minimum diameter of a subset of i nodes including the clause node, we observe

m(1) = 0, m(2) = b, m(3) = b+1,
m(4) = 2b, m(5) = 2b+1, m(6) = 2b+2 ...

It can be verified that to achieve minimum $\Sigma_i\ (n_i\text{-}1)d_i$, the partition should take the clause node and 3 nodes in a circuit into a subset with diameter 2b. This corresponds to assign at least one of the 3 literals in a clause to be true.

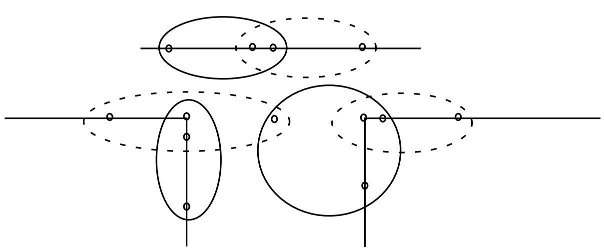

Fig.3 clause structure in Manhattan plane
(one of the three literals in the clause has to be assigned true to let the clause node in a cluster with diameter 2b)

Because we are in a rectilinear plane, the circuits have to cross each other. We need to carefully design the crossings such that the crossings won't change the optimum partitions of each circuit. We let the crossings take place only at two edges with length b(Fig.4). Denoting x(i) as the minimum diameter of a subset of i nodes near this crossing, we see

x(1) = 0, x(2) = 1, x(3) = b+1,
x(4) = b+2, x(5) = b+2, x(6) = b+2 ...

It can be verified that to achieve the minimum $\Sigma_i\ (n_i\text{-}1)d_i$, the six nodes near the crossing should be partitioned into two subsets of three nodes. There are two possible partitions which are essentially equivalent(Fig.4) since if we get an optimum partition containing one of the partitions, we can transform it into another partition, without changing the cost function and partitions of other nodes. So such a crossing structure does not impact the partition of each circuit.

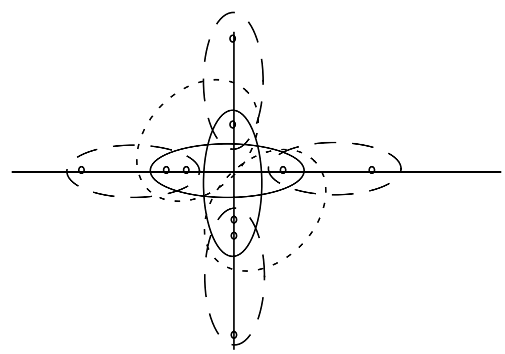

Fig.4 crossing of two circuits in Manhattan plane
(dashed and solid lines give two partitions corresponding to two different truth assignments, dotted and solid lines give two equivalent partitions corresponding to one truth assignment)

In summary, we construct v circuits corresponding to v variables in 3-SAT problem, m clause structure corresponding to m clauses in 3-SAT problem, and let the circuits crossing only at edges with length b. We then let $k = \Sigma\ S_i/3$, $B = 8mb + 3(k\text{-}m)(b+1)$, to solve the N_i-1 WSDKP problem.

Correctness:

If there is an assignment for the 3-SAT problem,then the corresponding partition over the nodes constructed as above has the cost of $B = 8mb + 3(k\text{-}m)(b+1)$. The corresponding partition problem is satisfied.

If there is a partition with cost $B \leq 8mb + 3(k\text{-}m)(b+1)$ for the partition problem, then

1. each circuit is partitioned into subsets of three nodes, and the crossings do not matter.
2. each clause node is in a subset of diamter 2b with other three nodes in a circuit, which corresponding to assign one of the three literals in the clause to be true.

Because such a partition yields a cost of B, and any deviation of such a partition can only increase the cost of $\Sigma_i\ (n_i\text{-}1)d_i$. So the corresponding 3-SAT problem is satisfied. Also the reduction is in polynomial time. q

Lemma 3: N_i-1 WSD K-way Partition Problem is NP-Complete in Euclidean plane.

Proof:
The proof is similar to that in Manhattan plane. (skipped due to lack of space) q
Lemma 4: N_i-1 WSD K-way Partition Problem is NP-Complete in diagonal plane.
Proof:
(skipped due to lack of space) q
Lemma 5: PBT_2 is NP-complete in Manhattan , Euclidean and diagonal plane.
Proof:
We show N_i-1 Weighted-Sum-Diameters K-way Partition Problem $\leq_p$ PBT_2. That is, we show if there is a polynomial algorithm to solve PBT_2, then we can construct a polynomial algorithm to solve the N_i-1 WSD K-way Partition Problem. Since N_i-1 WSD K-way Partition Problem is NP-complete, so is PBT_2.
Construct & Correctness:
Given a MSD K-way Partition Problem of $\Sigma_i D(C_i) \leq B$, construct a PBT_2 optimization problem of $Min(\Sigma_i D(C_i) + KC)$. Get k'(C), which corresponds to the optimum solution of PBT_2 problem for a fixed C. k'(C=0) = |V| = N, k'(C=∞) = 1. So a binary search is performed to find C_k such that $k'(C_k) = K$. The binary search takes O(lgN) time. Setting $B' = B + KC_k$, $C = C_k$, solve the PBT_2 Problem of $\Sigma_i D(C_i) + KC \leq B'$. Then the MSD K-way Partition Problem has a solution if and only if the PBT_2 problem so contructed has a solution. The time complexity is O(lgN) times the polynomial time complexity of PBT_2 problem, so is also polynomial. q
Lemma 6: PBT = PBT_n, where n = |V|.
Proof:
Since we allow the intermediate nodes coincident with each other, a PBT_n problem with large enough n has the same effect as the PBT problem without n constraint. And n=|V| is large enough. q
Theorem: PBT is NP-complete in Manhattan , Euclidean and diagonal plane.
Proof:
From Lemma 1-6. We have $PBT_h \leq_p PBT_{h+1}$. Since PBT_2 is NP-Complete, so is PBT_h for all h>2. So is PBT_n. Since PBT = PBT_n, so PBT is NP-Complete. q

4. Approximation Scheme

Due to the work of Sanjeev Arora and other people[11][12], many geometric problems including TSP, k-median and so on have been found to be approximated in $N^{O(1+c)}$ time to achieve (1+1/c) OPTIMUM. Applying their method, we find PBT problem with path-length delay model can be approximated in $N^{O(1+clgN)}$ time with (1+1/c) OPTIMUM.
Their contribution is that they find an (1+lgL/m) OPTIMUM solution can be found for a subproblem in any square with edge length L, and m portals on each edge. Portals are points only at which a approximation solution (such as a k-median connection) can cross the square boundary. A dynamic programming scheme is then implemented.A look-up table is built with each entry representing the best subproblem solution under condition of a possible location of the k medians inside and outside the square. The number of entries T is greatly reduced by introducing the portals, as $T = k(4m(3^{4m}))^2$. The look-up table is constructed bottom-up, by enumerating the look-up tables of the 4 children squares at the lower level. So the time to build each look-up table with T entries is T^5. The number of squares, hence the number of the look-up tables is O(NlgN). So the total time is $O((NlgN)T^5)$. To achieve (1+1/c)OPTIMUM, let m = clgL. Then the time spent is $N^{O(1)}L^{O(c)}$. Assuming L is propotional to N, it becomes $N^{O(1+c)}$.

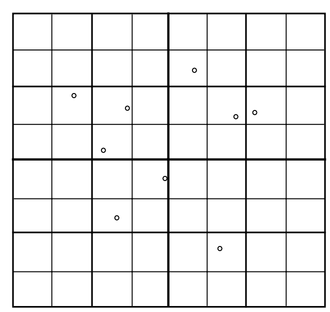

Fig.5 a geometric problem partitioned into subproblems

An instance of any square in k-median problem has the following inputs:
1. a nonempty square,
2. the number of medians inside the box, an integer f in [0,k],
3. If f 0, an assignment "inside" of numbers[1...4m] to the portals of the box representing the distance to the closest median inside the box,
4. If f < k, an assignment "closest" of numbers [1...4m] to the portals of the box representing the distance to the closest median outside the box.
By enumerating these inputs, we get the number of entries of the look-up table for this box, which is $T = k(4m(3^{4m}))^2$.
Applying this method, we get an approximation scheme for PBT problem with path-length delay model

similarly. While for an instance of a square in PBT problem, the inputs are:

1. a nonempty square,
2. the number f_i of nodes in set P with level i inside the box, for each i from 1 to the level J of the PBT,
3. If f_i 0, an assignment "inside(i)" of numbers[1...4m] to the portals of the box representing the distance to the closest node in set P with level i inside the box, for each i from 1 to the level J of the PBT,
4. If $f_i < k$, an assignment "closest(i)" of numbers [1...4m] to the portals of the box representing the distance to the closest node in set P with level i outside the box, for each i from 1 to the level J of the PBT.

So the number of entries of the look-up table for this box is $T = (f_i(4m(3^{4m}))^2)^J$ where $f_i = O(N)$, $J = O(\lg N)$. Then the total time spent is $(N\lg N)T^5 = N^{O(1+c\lg N)}$.

5. Conclusion

We prove PBT problem in Manhattan, Euclidean and diagonal plane is NP-complete, reveal the computational complexity of min cost zero-skew clock routing tree construction problem. While for path-length delay model, due to the planary property of the problem, an approximation scheme with time complexity of $N^{O(1+c\lg N)}$ exists.

References:

[1] R.S.Tsay, "An Exact Zero-Skew Clock Routing Algorithm", IEEE trans. on Computer Aided Design, pp.242-249, Feb., 1993

[2] Michael K.Gowan, Larry L.Biro and Daniel B.Jackson, "Power Consideration in the Design of the Alpha 21264 Microprocessor", 35th Proc. DAC, pp.726, 1998

[3] A.L.Fisher and H.T.Kung, "Synchronous large systolic arrays", Proc. SPIE,pp.44-52, 1982

[4] M.A.B.Jackson, A.Srinivasan and E.S.Kuh, "Clock routing for high-performance IC's", Proc. DAC, pp.573-579, June 1990

[5] Qing Zhu and Wayne Dai, "Perfect-balance Planar Clock Routing with Minimal Path-length", Proc. ICCAD, pp.473-476, Nov. 1992

[6] A.Kahng, J.Cong and G.Robins, "High-Performance clock routing based on recursive geometric matching", Proc. DAC, pp322-327, June 1991

[7] Masato Edahiro, "A Clustering-Based Optimization Algorithm in Zero-Skew Routings", 30th DAC, pp.612-616, 1993

[8] Andrew Kahng and Gabriel Robins, "On optimal interconnections for VLSI", Kluwer Publication, pp.147

[9] N.Megiddo and K.J.Supowit, "On the Complexity of Some Common Geometric Location Problems", Siam Journal of Computing, Vol.13, No.1, pp.182-196, 1984

[10] Michael R.Garey and David S.Johnson, "Computers and Intractability, A Guide to the Theory of NP-Completeness", W.H.Freeman, San Francisco, 1979

[11] Sanjeev Arora, "Polynomial-time Approximation Schemes for Euclidean TSP and other Geometric Problems", to appear in Journal of the ACM (URL:http://cs.princeton.edu/~arora/publist.html)

[12] Sanjeev Arora, Prabhakar Raghavan and Satish Rao, "Approximation Schemes for Euclidean k-medians and related problems", ACM STOC'98 (URL:http://cs.princeton.edu/~arora/publist.html)