Constrained Routing Problem

Pawan Kumar Patel
Department of Computer Science and Engineering
Indian Institute of Technology Kanpur, India

Rohit Kumar
Department of Electrical Engineering
Indian Institute of Technology (BHU) Varanasi, India

Nikita Gulati
Department of Computer Science and Engineering
Lucknow Model Institute of Technology Lucknow, India

ABSTRACT

In this paper, we have worked on a problem in the domain of graph theory and geometry .Objective of our problem is to find out the shortest path with forbidden pairs in a graphs. Given a graph G=(V, E) and set of pairs $P=\{a_{i,} b_i|a_i \in V\wedge b_i \in V\}$, we have to find out the shortest path between two given vertices s and t, s.t. a_i b_i both do not occur on the path for any i. We reduce SAT to this problem and thus claim that this problem is NP-hard.

Keywords
Shortest path with forbidden pairs, NP-hard, Computational geometry, Graph theory.

1. INTRODUCTION

Technique to identify NP-hard problems and to provide a reasonable approximation for them has been one of the most active area of algorithmic research. It involves a rich mathematical theory and promises a profound practical impact [1] [4] [7] [8] [9] .Graph theory and computational geometry are two domains offering many such open challenging problems. In this paper we have taken one problem and give mathematical proof to claim that to find shortest path with forbidden pairs is NP-hard.

Given a graph G=(V, E) with a set of pairs of vertices, called forbidden pairs, and two special vertices s and t. The goal is to find a shortest path between s and t such that at most one vertex is involved from each pair in this path.

One of applications of this problem is as follows. A message packet routed through internet passes through several intranets belonging to different domains. Suppose each domain perceives threat from a packet which has traveled through enemy territory. Thus it may not allow such packets to travel through their intranets, Then the routers will have a challenge to find an optimum path which does not pass through domains of any two enemy countries.

2. FINDING A PATH SUBJECT TO FORBIDDEN PAIRS IS NP-HARD

2.1 Problem

Let G=(V, E) is an undirected graph and $F=\{(x_{1,} x_2), (x_3, x_4), \ldots\ldots(x_{2k-1}, x_{2k})\}$ is a collection of pairs of vertices of G, called forbidden-pairs, such that $x_i \pm x_j$ for all $i \neq j$. In addision s and t are two specific vertices in G. Find a shortest path P in G connecting s and t subject to the forbidden-pair condition, i.e.for all $r \in [1,k]$, P passes through at most one of x_{2r-1} and x_{2r}. This is called forbidden pair shortest path problem (FPSP).

It will be shown that this problem is NP_hard by reducing CNF expression X= $(x_1 +x_2 + x'_3 + x_4)$. $(x'_2 +x_3 + x'_4)$. $(x'_1 + x_3 + x_4)$. $(x'_{1\,+}+ x'_2 + x'_4)$. It will be shpwn that this expression is satisfiable if and only if there is a path subject to the forbidden-pair condition between s and t in the graph of figure 1 where forbidden pairs are P= { $(a_1, b_1), (a_2, b_2), (c_1, d_1), (c_2, d_2), (e_{1,} f_1)_, (e_2, f_2), (g_1, h_1), (g_2, h_3), (g_3, h_2), (g_4, h_4)$}.

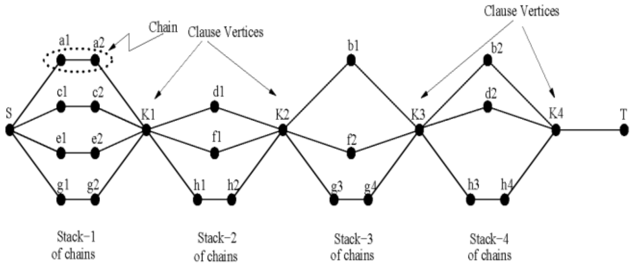

Fig 1: Graph constructed from Boolean expression

To relate the feasible paths to the Boolean value-assignment associate $\{a_1, a_2\}$ to x_1, $\{b_1, b_2\}$ to x'_1, $\{c_1, c_2\}$, to x_2, $\{d_1, d_2\}$ to x'_2, $\{f_1, f_2\}$ to x_3, $\{e_1, e2\}$ to x'_3, $\{g_1, g_2, g_3, g_4\}$ to x_4 and $\{ h_1, h_2,h_3,h_4\}$ to x'_4. Observe that in the figure 1, the stack of chains, from top to bottom, is associated with literals x_1, x_2, x'_3, x_4 respectively. These constitute the clause $C_{1.}$

Similarly the stack immediately to the left of k_j, is related to the literals of C_i for all i.

If path goes through any g_i, then we assign 1 to x_4 and 0 to x'_4. Similarly if it passes through any h_i then assign 1 to x'_4 and 0 to x_4. We will justify that the path will never pass through the vertices of both the sets, $\{g_1, g_2, g_3, g_4\}$ and $\{h_1, h_2, h_3, h_4\}$. Therefore above assignment is well defined. Similarly assign the value to other variables. If the path does not pass through the vertices of the set of a vertex or its complement, then choose the assignment randomly for that variable.

Suppose the path passes through g_1, g_2, then the forbidden pair condition requires that it cannot pass through h_1 or $h_{3.}$ From the structure of the graph, it evident that the path also cannot pass through h_2 and h_4 either. Similarly if the path passes through g_3, g_4, then again it can not pass through h_1, h_2, h_3, or $h_{4.}$ Due to symmetry we can see that if it pass through any h vertex then it can not pass through any g vertex. Thus if x_4 is assigned 1, then x'_4 can not be assigned 1 (consequently it will be assigned 0). Similarly if x'_4 is assigned 1, then x_4 will get value 0. Similar observations can be made for other variables. This shows that a path gives a valid value assignment if the variables associated with the vertices on the path are assigned 1.

The path reaches vertex k_i from s only by passing through one of the chains of the stack to its left. Thus one of the literals of C_i must be assigned value 1. Therefore we see that if there exists a feasible path from s to t, then the associated value assignment satisfies the expression X.

Next we show that every satisfying assignment has at least one associated path. Consider a value assignment which satisfies the expression. Let V' be a set of vertices including s, t, k_1, k_2, k_3, k_4 and those vertices among a_i, b_{i},………., h_i which are associated with the literal having value 1. Observe that V' cannot have both members of any forbidden pair. The value assignment satisfies the expression therefore at least one chain in each stack is contained in V'. Thus the induced grapg on V' must be connected. Since it contains both, s and t, there exists at least one path connecting the two and it is feasible.

These observations lead to the following result. It shows that X has a satisfying assignment iff there is a at least one feasible path between s and t in the graph subject to the forbidden pairs P. We also notice that the total number of vertices in the graph is at most forbidden pairs P. We also note that the total number of vertices in the graph is at most n(m-1) + mm + 2 where m is the total number of clauses and n is the total number of literals occurrences.

3. FORMAL PROOF

3.1 Reduction of SAT to FPFP

Let X be a CNF expression containing K clauses. If a variable occurs in the expression but its complement does not, then the clauses containing this variable can trivially be satisfied without affecting other clauses. Similarly if the complement of a variable occurs without its variable then again the corresponding clauses can be satisfied trivially. Thus we can drop such clauses from the expression. Henceforth we will assume that no variable (respectively, complement of a variable) occurs without its complement (respectively. Its variable) in X.

Let C_1, C_2,............,C_k denote the clauses and x_1, x_2,……denote the variables of the expression. Define a graph G=(V, E) as follows. The vertex set is $\{ S = k_{0,}\, k_1,\ldots\ldots,k_k, T=k_{k+1}\} \cup \{a_{ipj} : x_p \in C_i, x'_p \in C_j\} \cup \{ b_{jpi} : x'_p \in C_j, x_p \in C_i\}$. Suppose x_p occurs in clauses C_{j1},………,C_{jmp} with $j_1 < j_2 <\ldots\ldots< j_{mp}$ and x'_p occurs in $C_{j'1}$, ……….,$C_{j'mp}$ where $j'_1 < j'_2 <\ldots\ldots< j'_{np}$. Then define these edges: (i) $(a_{JupJv}, a_{JupJ'v+1})$ for every u, v (ii) $(b_{JvpJu}, b_{Jvpj'u+1})$

For every u, v (iii) $(k_{Ju-1}, a_{JupJ'1})$ for every u (iv) $(k_{j'v-1}, b_{J'cPJ1})$ for every v (v) $(a_{JuPnp,}\, k_{Ju})$ for every u (vi) $(b_{J'vPmp,}\, k_{J'v})$ for every v. Similar edges for each p. Finally include the edge (k_K, k_{K+1}). Figure 2 shows the graph for

$X = (x_1 + x_2 + x'_3 + x_4) . (x'_2 + x_3 + x'_4) . (x'_1 + x_3 + x_4). (x'_1 + x'_2 + x'_4)$ based on the labeling convention described above.

The forbidden pair-par problem constructed from SAT problem includes this graph with the pairs $F = \{ (a_{ypz}, b_{zpy})\}_{yzp}$[2] [5] [6] [7] [8] [9]. To prove the NP-hardness of FPFP we will show that X is satisfiable iff this G has a feasible path between s and t.

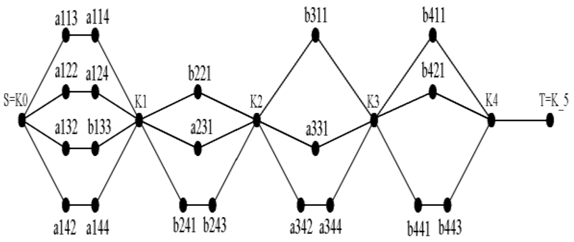

Fig 2: Example of labeling convention

(only if) Consider a satisfying assignment for X. Define a vertex set $V' = \{k_0, k_1,\ldots\ldots,k_K, k_{K+1}\} \cup \{a_{ypz} : x'_p = 1\}$. For each k_i, either there is some $x_p \in C_{Ki}$ with value 1 or there is some $x'_p \in C_{Ki}$ with value 1. In former case, $\{ a_{ipj'1}, a_{ipj'2,} \ldots\ldots, a_{ipj'np} \}$ are in V', in the later case , $\{ b_{ipj'1}, b_{ipj'2,} \ldots\ldots, b_{ipj'mp} \}$ V'. In either case there is a path from k_{i_1} to k_i in the subgraph induced by V', $G(V')$. Therefore this subgraph is connected. This implies that there is a path from k_0 to k_{K+1} in $G(v')$. It is not difficult to see that V' does not have both vertices of any forbidden pair since if a variable (or complement) has value 1, then its complement (resp, variable) has value 0.

(If) We are given a path from k_0 to k_{K+1}=t satisfying the forbidden-pair constraint. Each k_i, for $1\leq I \leq K$, is a cut-vertex and s and t are in the different components of $G\{V\text{-}\{k_i\}\}$ therefore the path must pass through all k_i. From the structure of the graph it is obvious that the path must pass through a chain in each stack.

Consider the stack adjacent to k_y on the left side (s-side). Suppose the path passes through a chain a_{yp*}. We will show that the path will not pass through any vertex b_{*p*} in the graph. Suppose the chan has nodes : $a_{ypj'1}$, : $a_{ypj'2}$……………: $a_{ypj'np.}$ Thus b_{zpy} must be defined only for $z \in \{j'_1,\ldots\ldots\ldots j'_{np}\}$. From forbidden-pair constraint the path cannot pass through b_{zpy} for each z in the set. The chain vertices have degree 2 so the path cannot pass through any vertex b_{zp*} for each $z \in \{j'_1,\ldots\ldots\ldots j'_{np}\}$, i.e., any vertex b_{*p*}. Similar argument shows that if the path pass through any vertex b_{zpy} then it cannot pass through any vertex $a_{*p*.}$ In other words, for each p, either all b_{*p*} are avoided by the path or all a_{*p*} are avoided by it.

Define a Boolean value assignment $x_p = 1$ if the path passes through a_{ypz} for some y,z. Similarly assign $x'_p = 1$ if any b_{zpy} is on the path. Set all the complementary literals to zero. If the path does not pass through any a_{ypz} or b_{zpy} for any y, z then the set the value of x_p arbitrarily.

In the light of conclusion reached in the previous paragraph it will never happen that a variable and it complement both are assigned 1 (0). Thus this value assignment is valid [3].

To see that this value assignment satisfies the Boolean expression X, observe that for each clause C_i there is a stack of vertices with labels a_k….or b_k…. coreesponding to the literals of that clause. The path must pass through one of these chains so the literal associated with that chain must be assigned value 1. Thus the value assignment must satisfy X.

4. CONCLUSION

In this paper we thoroughly concludes that every constraint routing problem is Np-hard problem through conversion of this problem into well known NP-hard problem (SAT). Several well known technique available to deal NP- Hard problem(SAT) makes this highly application based problem popular due to easy mapping from constrained routing problem into SAT problem. Due to wider application of this problem in communication and network where security issue is at most concerned carries a lot of advantages to study further to give optimal solution of this problem with help of approximation algorithms.

5. REFERENCES

[1] P. Raghavendra. Optimal algorithms and inapproximability results for every csp? In Proceedings of the 40th ACM Symposium on the Theory of Computing, pages 245{254, New York, NY, USA, 2008. ACM.

[2] Y. Zhao, X. Deng, C. H. Lee and H. Zhu, (2 + f(n))-SAT and its properties, Discrete Applied Mathematics 136 (2004), 3–11

[3] S. Aaronson. Is P versus NP formally independent? Bulletin of the European Association for Theoretical Computer Science, 81, Oct. 2003.

[4] R. Impagliazzo, R. Paturi, and F. Zane, Which Problems Have Strongly Exponential Complexity?, Journal of Computer and System Sciences 63-4, (2001), pp. 512-530.

[5] J. Gramm, and R. Niedermeier, Faster exact solutions for Max-2-Sat, in Proceedings of the 4th Italian Conference on Algorithms and Complexity (CIAC), Lecture Notes in Computer Science 1767, (2000), pp. 174-186.

[6] B. Borchers and J. Furman, A two-phase exact algorithm for Max-Sat and weighted Max-Sat problems, J. Combinatorial Optimization 2, (1999), pp. 465-474.

[7] T. Asano, and D. P. Williamson, Improved Approximation Algorithms for Max-Sat, in Proceedings of the 11th ACM-SIAM Symposium on Discrete Algorithms (SODA), (2000),

[8] N. Bansal and V. Raman, Upper bounds for Max-Sat further improved, in Proceedings of the 10th International Symposium on Algorithms and Computation (ISAAC), Lecture Notes in Computer Science 1741, (1999), pp. 247-258.

[9] Battiti and M. Protasi, Reactive research, a history base heuristic for Max-Sat, J.Exper. Algorithmics 2, No. 2, (1997).