Eulerian disjoint paths problem in grid graphs is **NP**-complete*

Dániel Marx†

12th December 2003

Abstract

We show that the edge disjoint paths problem is **NP**-complete in directed or undirected rectangular grids, even if the union $G + H$ of the supply and the demand graph is Eulerian.

Keywords: disjoint paths, grids, **NP**-completeness

1 Introduction

Disjoint paths problems arise naturally in practical applications such as network routing and VLSI-design. The problem is also interesting from the theoretic point of view: there are several beautiful good characterization theorems for some restricted cases. The restriction to planar graphs, and in particular to planar grid graphs is both of practical and theoretical interest. Here we prove the **NP**-completeness of a planar case of the problem, settling an open question of Vygen [6]. This complements the good characterization theorem of Okamura and Seymour.

In the disjoint paths problem we are given a graph G and a set of source–destination pairs (s_1, t_1), (s_2, t_2), ..., (s_k, t_k) called the *terminals*, and we have to find k disjoint paths P_1, ..., P_k such that path P_i connects vertex s_i to vertex t_i. There are four basic variants of the problem: the graph can be directed or undirected, and we can require edge disjoint or vertex disjoint paths. The problem is often described in terms of a supply graph and a demand graph, as follows:

Disjoint Paths

Input: The *supply graph* G and the *demand graph* H on the same set of vertices.

Task: Find a path P_e in G for each $e \in E(H)$ such that these paths are pairwise disjoint and path P_e together with edge e forms a circuit.

The graphs G and H can have parallel edges but no loops. For vertex disjoint paths we allow their endpoints to be the same. In the directed version of the problem both

*Research is supported by grants OTKA 44733, 42559 and 42706 of the Hungarian National Science Fund

†Department of Computer Science and Information Theory, Budapest University of Technology and Economics, H-1521 Budapest, Hungary. dmarx@cs.bme.hu

G and H are directed. With a slight abuse of terminology, we say in the directed case that a demand $\overrightarrow{uv} \in H$ starts in v and ends in u (since the directed path satisfying this demand starts in v and ends in u). Moreover, given a solution of the disjoint paths problem, we identify a demand with the path satisfying it. That is, we say that "demand α uses supply edge e" instead of "the path satisfying demand α uses edge e". An undirected graph is called *Eulerian* if every vertex has even degree, and a directed graph is Eulerian if the indegree equals the outdegree at every vertex.

The disjoint paths problem and its variants were intensively studied, for an overview see [2, 6]. In particular, all four variants of the problem (directed/undirected, edge disjoint/vertex disjoint) are **NP**-complete, even when G is planar. In this paper we consider only the (directed and undirected) edge disjoint paths problem in the grid, thus henceforth disjoint means edge disjoint.

If there exist disjoint paths in $G(V, E)$ with the given endpoints, then every cut $(V', V \setminus V')$ has to contain at least as many edges from G as from H, otherwise there would be more demands crossing this cut than edges connecting V' and $V \setminus V'$. We say that the *cut criterion* holds for G and H if this is true for every cut $(V', V \setminus V')$. In general, the cut criterion is only a necessary condition, but in an important special case it is also sufficient:

Theorem 1 (Okamura and Seymour, 1981, [5]). *Assume that G is planar, undirected, $G + H$ is Eulerian, and every edge of H lies on the outer face of G. The edge disjoint paths problem has a solution if and only if the cut criterion is satisfied.*

A graph is a *grid graph* if it is a finite subgraph of the rectangular grid. A directed grid graph is a grid graph with the horizontal edges directed to the right and the vertical edges directed to the bottom. Clearly, every directed grid graph is acyclic. A *rectangle* is a grid graph with $n \times m$ nodes such that $v_{i,j}$ $(1 \leq i \leq n,\ 1 \leq j \leq m)$ is connected to $v_{i',j'}$ if and only if $|i - i'| = 1$ and $j = j'$, or $i = i'$ and $|j - j'| = 1$. The study of grid and rectangle graphs is motivated by applications in VLSI-layout.

The special case of Theorem 1 when G is a rectangle is investigated in [1]. The requirement that the edges of H lie on the outer face of G cannot be dropped even in this restricted case: Figure 1 shows an example where $G + H$ is Eulerian, the cut criterion holds, but the terminals cannot be connected by edge disjoint paths (it is left to the reader to verify these claims). In Section 2, we prove that the edge disjoint paths problem is **NP**-complete on rectangles even if $G + H$ is Eulerian. This answers an open question of Vygen [6]. Moreover, this also implies that (unless **coNP** = **NP**) a generalization of Theorem 1 cannot give a good characterization to the case when we drop the requirement that the terminals have to lie on the outer face.

There are several good characterization theorems in the literature [2, 6] for the case when G is planar, $G + H$ is Eulerian, and some additional constraint holds (as in Theorem 1). Previously no **NP**-completeness result was known for G planar and $G+H$ Eulerian. To the best of our knowledge, the only negative result for $G + H$ Eulerian is the theorem of Vygen [7] stating that the disjoint paths problem is **NP**-complete if $G + H$ is Eulerian, and G is an undirected (nonplanar) graph or a DAG.

In the directed case, Vygen proved that the edge disjoint paths problem is **NP**-complete even if the supply graph G is planar and acyclic [7] or even if G is a directed rectangle [6], and asked whether the problem remains **NP**-complete with the additional constraint that the graph $G + H$ is Eulerian. We settle this question by proving that the problem, similarly to the undirected version, is indeed **NP**-complete.

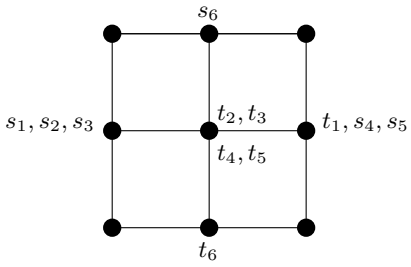

Figure 1: Example showing that Theorem 1 does not hold when the terminals do not have to lie on the outer face.

2 The reduction

In this section we prove that the edge disjoint paths problem on directed and undirected rectangle graphs remains **NP**-complete even in the restricted case when $G + H$ is Eulerian. First we prove that the problem is **NP**-complete on directed grid graphs with $G + H$ Eulerian. Using standard techniques, this result is extended to rectangle graphs and undirected graphs.

The following observation will be useful:

Lemma 2. *In the directed disjoint paths problem, if $G + H$ is Eulerian and G is acyclic, then every solution uses all the edges of G.*

Proof. Assume that a solution is given. Take a demand edge of H and delete from $G + H$ the directed circuit formed by the demand edge and its path in the solution. Continue this until the remaining graph contains no demand edges, then it is a subgraph of G. Since we deleted only directed circuits, it remains Eulerian, but the only Eulerian subgraph of the acyclic graph G is the empty graph with no edges, thus the solution used all the edges. □

Proving the **NP**-completeness of a planar problem is usually done in one of two ways: either the reduction is from a planar problem (such as planar SAT, planar independent set etc.) or the reduction constructs a planar instance by locally replacing crossings with copies of some crossover gadget (as in [3] for planar graph coloring). Our reduction is none of these two types: there are crossings, but the global structure of the construction ensures that the crossings "behave nicely". This resembles the way [7] proves the **NP**-completeness of the disjoint paths problem on planar DAGs.

Theorem 3. *The edge disjoint paths problem is* **NP***-complete on directed grid graphs, even if $G + H$ is Eulerian.*

Proof. The proof is by polynomial reduction from a restricted case of 1-in-3 SAT, where a formula is given in conjunctive normal form, and our task is to find a variable assignment such that in every clause of the formula, exactly one of the three literals is true. In *monotone* 1-in-3 SAT every literal is positive (not negated), and in the *cubic* version of the problem every variable occurs exactly three times. In [4] it is shown that monotone, cubic 1-in-3 SAT is **NP**-complete.

Let n be the number of variables in the given monotone, cubic 1-in-3 SAT formula, this obviously equals the number of clauses. It can be assumed that every clause contains three different literals. The reduction is of the component design type: we

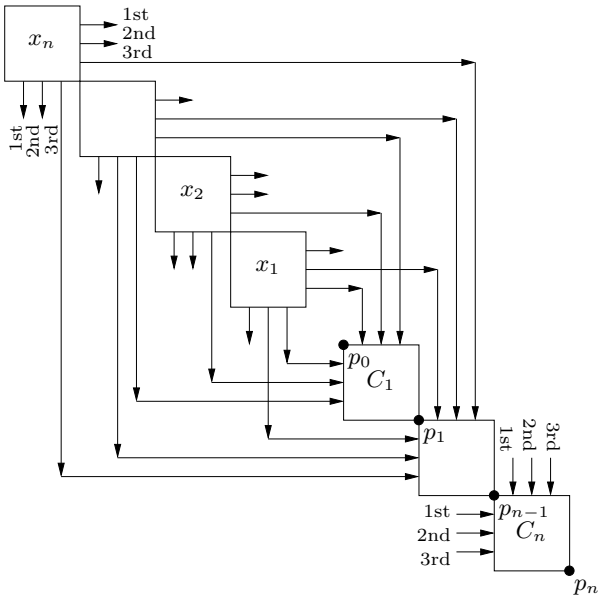

Figure 2: Overview of the reduction. The variable setting gadgets are on the left, three paths leave each of them to the right and to the bottom. These paths lead to the satisfaction testing gadgets below.

construct variable setting gadgets and satisfaction testing gadgets, and connect them in such a way that the disjoint paths problem has a solution if and only if the given formula is satisfiable (in 1-in-3 sense). The constructed graph G is a grid graph, and the construction ensures that $G + H$ is Eulerian in the resulting instance.

First we present how the gadgets are connected, the structure of the gadgets itself will be described later. Going diagonally from top left to bottom right, place a sequence of n copies of the variable setting gadget. The component corresponding to x_n is in the top left corner. Continue this sequence by n copies of the satisfaction testing gadget (see Figure 2). Denote by p_t the lower right vertex of the component corresponding to the tth clause, and let p_0 be the top left vertex of the component of the first clause. Three paths leave each variable gadget to the right and three to the bottom, they will be called the *right exits* and the *lower exits* of the gadget. The exits are numbered, the topmost right exit is the first right exit, and the leftmost lower exit is the first lower exit. Similarly, the satisfaction testing gadgets have three *upper entries* (the first is the leftmost) and three *left entries* (the first is the topmost). Assume that the literals in a clause are sorted, the variable of the first literal has the smallest index, i.e., as in the clause $(x_1 \vee x_2 \vee x_7)$. The occurrences of a variable are numbered in such a way that the first occurrence of the variable is in the clause with the *largest* index.

The components are connected as follows. If the ith occurrence ($i = 1, 2, 3$) of variable x_s is the jth literal ($j = 1, 2, 3$) in clause C_t, then connect the ith right exit of the component of x_s to the jth upper entry of the component of C_t, and similarly with lower exits and left entries. Each connection is a path in the grid consisting of several directed edges. The connections are done by first going to the right (below) and then to below (right), there is only one turn in each connection. There will be exactly $6n$ demands: if variable x_s appears in clause C_t, then there are two demands that start

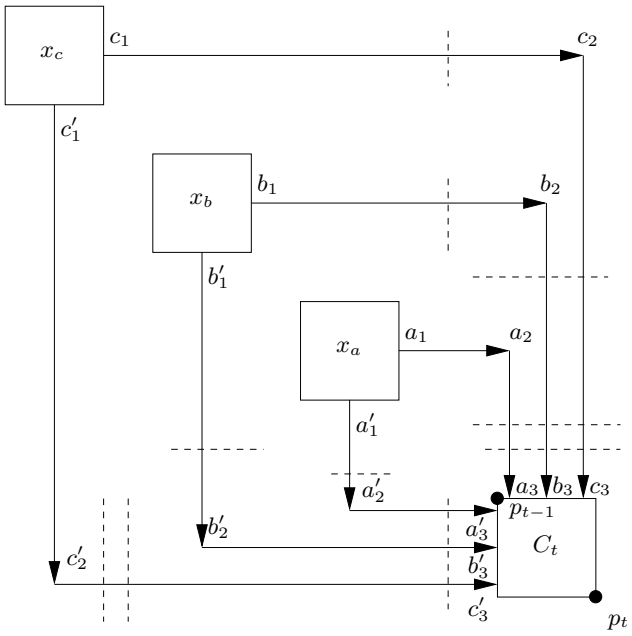

Figure 3: The paths entering a given satisfaction testing component.

from the component of x_s and end in the component of C_t. (The exact location of the start and end vertices of the demands will be defined later.)

The connections described above can cross each other at a vertex, there may be several such crossings in the resulting grid graph. Given a solution of the disjoint paths problem, we call a vertex a *bad crossing*, if the demand entering this vertex from the left leaves to the bottom, and the demand entering from above leaves to the right. (Note that by Lemma 2, exactly two demands go through a crossing). We show by induction that a solution in this graph cannot contain a bad crossing. Clearly, there are no crossings to the left and above of the vertex p_0. Assume that there are no bad crossings to the left and above of the vertex p_{t-1}. Figure 3 shows the paths entering the component of clause C_t, the dashed lines show other possible paths that may cross these six paths. By the way the literals are ordered in the clause, the six paths entering a clause component do not cross each other (recall that the component of x_n is in the upper left corner). Furthermore, because of the way the occurrences of a variable are ordered, the paths leaving a variable component do not cross each other either.

By the induction hypotheses, the same demand goes through vertices c_1 and c_2, through vertices b_1 and b_2, through vertices a_1 and a_2, through vertices c'_1 and c'_2, and so on. For example, the demand going through c_1 can leave the path c_1c_2 only if there is a bad crossing on c_1c_2, but there are no bad crossings to the left and above of p_{t-1}. There are two demands that starts in the component of x_c and have C_t as destination. They cannot leave x_c both to the right: only one of them can reach C_t through the path from c_2 to c_3, and there is no other way of reaching C_t without a bad crossing to the left of p_{t-1}. Similarly, the two demands cannot leave both to the bottom. Thus exactly one of the demands going to C_t leaves to the right and the other to the bottom, furthermore, these demands leave x_c through c_1 and c'_1. By a similar argument, this also holds for the components of x_b and x_a. Clearly, the demand going

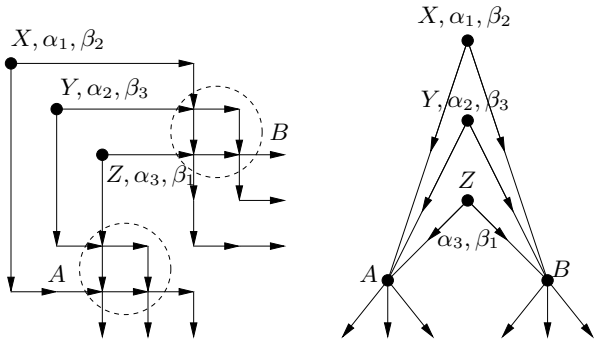

Figure 4: The variable setting gadget.

through c_1 and c_2 can reach C_t only through c_3, thus there are no bad crossings on the path from c_2 to c_3. The demand going through b_2 can reach C_t only through b_3 or c_3, but since c_3 is already used, only b_3 remains. Finally, the demand going through a_2 has to enter C_t in a_3. Therefore there are no bad crossings above C_t, and a similar argument shows that there are no bad crossings to the left of C_t. Thus there are no bad crossings to the left and above of p_t, which completes the induction.

Now we describe the gadgets used in the reduction. The *variable setting gadget* (Figure 4) has three output edges to the right, and three output edges to the bottom. On the right of the figure a simplified version of the gadget is shown, which is not a grid graph, just a planar DAG. The structure of the real gadget is the same, but in order to make it a grid graph some of the edges have to be twisted and the high-degree vertices A and B have to be split. We will show how the simplified version works, it is easy to show that the same holds for the real gadget.

Demands α_1, β_2 start in X, demands α_2, β_3 start in Y, and demands α_3, β_1 start in Z. The destination of demands α_i and β_i are in the clause component corresponding to the clause of the ith occurrence of the variable. We have seen that in every solution either α_i leaves the gadget to the right and β_i leaves to the bottom, or the opposite. However, more is true: either all of $\alpha_1, \alpha_2, \alpha_3$ leave to the right (through B) and $\beta_1, \beta_2, \beta_3$ leave to the bottom (through A), or the other way. To see this, first assume that α_1 uses $\overrightarrow{XA}$, then β_2 uses $\overrightarrow{XB}$. This implies that α_2 cannot go through B, thus α_2 uses $\overrightarrow{YA}$ and β_3 uses $\overrightarrow{YB}$. Demand α_3 cannot go through B, hence it uses $\overrightarrow{ZA}$ and β_1 uses $\overrightarrow{ZB}$. Thus $\alpha_1, \alpha_2, \alpha_3$ go through A, and $\beta_1, \beta_2, \beta_3$ go through B, what we had to show. By a similar argument, if α_1 uses $\overrightarrow{XB}$, we get that all three demands α_i go through B. Therefore in every solution of the disjoint paths problem, the component of x_s has two possible states: either the demands α_i leave to the bottom (we call this state 'true') or they leave to the right ('false'). Recall that if the demands α_i leave to the bottom, then they reach their respective clause components from the left, while if they leave to the right, then they reach the clause components from the top.

The *satisfaction testing gadget* and its simplified equivalent is shown on Figure 5. The three paths that enter K correspond to the three paths that enter the gadget from the left, while the paths entering L correspond to those entering from the top. The gadget contains the endpoints of six demands corresponding to the three variables. Demands γ_j and δ_j start in the variable component corresponding to the jth literal of the clause. More precisely, if the jth literal of clause C_t is the ith occurrence of variable

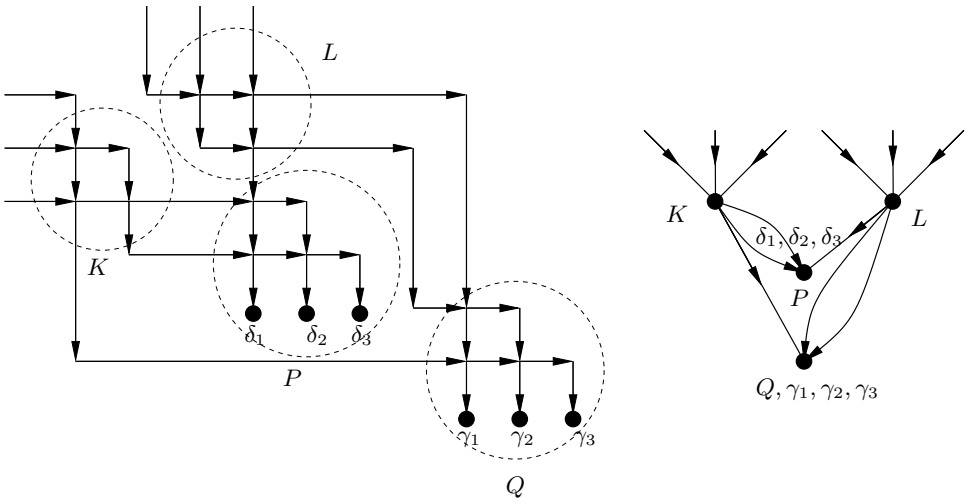

Figure 5: The satisfaction testing gadget.

x_s, then demand α_i starting in gadget x_s is the same as demand γ_j terminating in gadget C_t, and β_i is the same as δ_j.

Vertex P is the endpoint of the three demands δ_1, δ_2, δ_3, and vertex Q is the endpoint of the demands γ_1, γ_2, γ_3. We have seen that in every solution, exactly one of γ_j and δ_j leaves the variable component to the right, the other one leaves to the bottom, hence exactly one of them enters the clause component from the top, the other one enters from the left. Furthermore, there is exactly one j such that γ_j enters from the left and δ_j enters from the top, for the remaining two $j' \neq j$, demand $\gamma_{j'}$ enters from the top and demand $\delta_{j'}$ enters from the left. To see this, notice that from K only one demand can reach Q and only two demands can reach P. Thus the satisfaction testing gadget effectively forces that exactly one of the three variable gadgets is in the state 'true'.

It can be easily verified that $G + H$ is Eulerian in the constructed instance. Given a solution to the disjoint paths problem, we can find a satisfying assignment of the formula: assign to the variable x_s 'true' or 'false' depending on the state of the gadget corresponding to x_s. By the construction, every clause will be satisfied (in 1-in-3 sense). On the other hand, given a satisfying variable assignment, we can find a solution to the disjoint paths problem: the values of the variables determine how the demands leave the variable setting gadgets and this can be extended to the whole graph. □

It is noted in [6] that the disjoint paths problem is not easier in rectangle graphs than in general grid graphs: if we add a new edge $\overrightarrow{uv}$ to G and a new demand from u to v, then the new demand can reach v in the grid only using the new edge. Thus we can add new edges and demands until we get a full rectangle graph without changing the solvability of the instance. Clearly, $G + H$ remains Eulerian after adding the supply edge $\overrightarrow{uv}$ to G, and the demand edge $\overrightarrow{vu}$ to H.

Corollary 4. *The edge disjoint paths problem is* **NP***-complete on directed rectangle graphs, even if $G + H$ is Eulerian.*

A reduction from the directed case to the undirected one was described by Vygen:

Lemma 5 ([7]). *If (G, H) is an instance of the directed edge disjoint paths problem, where G is acyclic, $G + H$ is Eulerian, and the undirected graphs G', H' result from neglecting the orientation of G, H, then every solution of (G, H) is also the solution of (G', H') and vice versa.*

Combining Corollary 4 and Lemma 5, we obtain the following corollary, settling another open question from [6]:

Corollary 6. *The undirected edge disjoint paths problem is* **NP***-complete on rectangle graphs, even if $G + H$ is Eulerian.*

Acknowledgments

I'm grateful to Katalin Friedl for her help in preparing the paper. The useful comments of Judit Csima are also acknowledged.

References

[1] A. Frank. Disjoint paths in a rectilinear grid. *Combinatorica*, 2(4):361–371, 1982.

[2] A. Frank. Packing paths, circuits, and cuts—a survey. In *Paths, flows, and VLSI-layout (Bonn, 1988)*, pages 47–100. Springer, Berlin, 1990.

[3] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoret. Comput. Sci.*, 1(3):237–267, 1976.

[4] C. Moore and J. M. Robson. Hard tiling problems with simple tiles. *Discrete Comput. Geom.*, 26(4):573–590, 2001.

[5] H. Okamura and P. D. Seymour. Multicommodity flows in planar graphs. *J. Combin. Theory Ser. B*, 31(1):75–81, 1981.

[6] J. Vygen. Disjoint paths. Technical Report 94816, Research Institute for Discrete Mathemathics, University of Bonn, 1994.

[7] J. Vygen. NP-completeness of some edge-disjoint paths problems. *Discrete Appl. Math.*, 61(1):83–90, 1995.