Electronic Colloquium on Computational Complexity, Report No. 23 (2003)

On Spanning Cacti and Asymmetric TSP

Anna Palbom
Department of Numerical Analysis and Computer Science
Royal Institute of Technology
Stockholm, Sweden
E-mail: annap@nada.kth.se

February 12, 2003

Abstract

In an attempt to generalize Christofides algorithm for metric TSP to the asymmetric TSP with triangle inequality we have studied various properties of *directed spanning cacti.* In this paper we first observe that finding the TSP in a directed, weighted complete graph with triangle inequality is polynomial time equivalent to finding the minimum spanning cactus in the graph and then prove that it is NP-complete to determine whether a general unweighted digraph contains a directed spanning cactus. The proof is a reduction from Exact cover and may be of independent interest.

Keywords: directed cacti, spanning cacti, NP-complete, asymmetric TSP

1 Introduction

The Traveling Salesman Problem (TSP) is one of the most famous and well-studied NP-problems. It was proven NP-complete already by Karp [5]. This means that an efficient algorithm for TSP is highly unlikely; hence it is interesting to investigate algorithms that compute *approximate* solutions. However Sahni and Gonzalez [7] showed that in the case of general metrics it is NP-hard to find a tour with weight within polynomial factors of the optimum. When the metric is symmetric and constrained to satisfy the triangle inequality there exists a folklore 2-approximation algorithm, i.e., an algorithm that finds a tour no longer than two times the length of the optimal tour: Compute a minimum spanning tree and then transform it into a TSP tour. The best known approximation algorithm is a factor 3/2-approximation algorithm due to Christofides [1]. It constructs a TSP tour from a spanning tree and a matching.

The above algorithms apply only to symmetric distance functions. The asymmetric case is much less understood. The best known algorithm, invented by Frieze, Galbiati and Maffioli [3], approximates the TSP tour within a factor of $\log n$. In their paper, Frieze et al. pose the question regarding the optimality of their algorithm. This question is still one of the most intriguing open questions in the field of approximation algorithms. The by now twenty-year-old algorithm of Frieze et al. [3] is still the best known algorithm and there is only a miniscule lower bound: Papadimitriou and Vempala [6] recently proved that it is NP-hard to approximate the minimum TSP tour within a factor less than $220/219 - \epsilon$, for any constant $\epsilon > 0$.

Hence, any algorithm approximating the minimum TSP tour within a factor independent of the number of vertices n is of great interest to the community. In order to construct such an algorithm

ISSN 1433-8092

it is natural to try to generalize the ideas used by Christofides [1]. In particular, it seems fruitful to search for structures similar to that of a spanning tree in asymmetric graphs. One such structure is the spanning cactus:

Definition 1. *[8] A strongly connected, directed graph where each edge is contained in at most (and thus, in exactly) one directed cycle is called a* directed cactus.

Definition 2. *A* spanning, directed cactus *for a directed graph G is a subgraph of G that is a directed cactus and connects all vertices in G.*

Our first observation is that finding the minimum spanning cactus and the minimum TSP tour are polynomial time equivalent problems. They also have the same hardness of approximation. Therefore it can not be easier to find a minimum spanning cactus than a minimum TSP tour.

Theorem 1. *Finding a spanning cactus of minimum total edge weight in an asymmetric, weighted, complete graph where the weights obey the triangle inequality is polynomial time equivalent to finding the minimum TSP tour in the same graph. They also have the same hardness of approximation.*

We then consider cacti for their own sake. More precisely, we study the problem of finding a spanning cactus in a general, unweighted, directed graph and proved that it is NP-complete.

Definition 3. Spanning cactus problem (SCP)*: given a directed graph, decide if there is a spanning, directed cactus in the graph.*

Theorem 2. *SCP is NP-complete.*

This result indicates that cacti in asymmetric graphs are not useful as a tool in the construction of approximation algorithms for other NP-complete problems.

1.1 Previous work

In discrete mathematics cacti are an accepted graph structure and in undirected graphs they have been carefully studied. Finding the minimum cut in a graph is a well-known optimization problem. Here a cactus is a useful and simple representation of the minimum cuts in a graph (there can be many). Cacti for this purpose are used for example by Fleischer in [2].

Directed cacti are much less studied than the undirected ones. In 1994 Schaar [8] published a paper about Hamiltonian properties of directed graphs. He showed some results about graphs restricted to be directed cacti. As far as we know no one has shown anything about the complexity of finding a spanning cactus in a directed graph.

1.2 Notations and conventions

In a directed graph an edge from vertex A to vertex B is denoted AB, a path from A to B to C is denoted ABC and a cycle from A to B to C and back to A is denoted ABCA. Considered cycles are always *simple*.

When we look at subgraphs (such as gadgets) we use the term *cactus branch*. We define the term as:

Definition 4. *Suppose there is a spanning cactus S in a directed graph G. In a subgraph $H \subseteq G$ a* cactus branch *is the set of edges $\{e : e \in S \cap H\}$.*

2 Proof of Theorem 1

We first prove that the minimum spanning cactus in a graph is in fact also the minimum TSP tour and vice versa.

The TSP tour is a spanning cactus and therefore the weight of the minimum spanning cactus is less than or equal to the TSP tour's weight.

If we have the minimum spanning cactus it is possible to transform it into a TSP tour in the following way: Start in an arbitrary vertex, traverse the spanning cactus in the order of an Euler tour. If an edge goes to an already visited vertex replace the edge to the vertex and the next edge in the Euler tour with the edge short-cutting them. If the new edge goes to a visited vertex repeat until an unvisited vertex is found or to the end of the Euler tour. The triangle inequality gives that the weight of the short-cut edge is less than or equal to the combined weight of the original edges. The found TSP tour therefore has a weight less than or equal to the minimum spanning cactus's weight.

Secondly, we prove that TSP can be approximated within c if and only if the size of the spanning cactus can be approximated within c. Every TSP tour is a spanning cactus and hence a c-approximation algorithm for TSP approximates the minimum spanning cactus within the same ratio. Conversely, a c-approximation algorithm for the minimum spanning cactus can be used to construct a c-approximate TSP tour by the construction outlined in the previous paragraph.

3 Proof that SCP is NP-complete

We will first show that SCP is in NP and then reduce Exact cover (which is a NP-complete problem [5]) to SCP.

Lemma 1. *SCP is in NP.*

Proof. The definition of an NP-problem is that if the problem has a solution there is a witness which convinces a verifier that the problem is solvable. It should be possible to check the witness in polynomial time.

Our witness of SCP is the spanning cactus itself and the edges are given in the order of an Euler tour. We will prove that we can check if the Euler tour is a spanning cactus in polynomial time.

If every vertex is in the Euler tour the subgraph is strongly connected.

To make sure that every edge is in at most one cycle, traverse the edges in the given order. Then push every visited vertex on a stack. If we come to an already visited vertex, pop all vertices above it (but not the vertex itself). Continue until all edges of the tour have been visited. If we come to an already visited vertex, which is not on the stack, the test does not accept the graph as a cactus, otherwise it does.

Thus we can check the witness of a spanning cactus in polynomial time. □

Remark: It is enough that the witness gives the edges of the cactus. The Euler tour can be found in polynomial time.

3.1 Reducing Exact Cover to SCP

Exact cover is a well-known NP-complete problem [5]. By reducing Exact cover to SCP we will show that SCP is NP-complete as well.

Definition 5. Exact cover problem*: given a family $F = \{S_1, S_2, ..., S_n\}$ of subsets of a set $U = \{u_1, u_2, ..., u_m\}$. Is there a subset $C \subseteq F$ such that each $u_i \in U$ is in exactly one of the subsets $S_j \in C$?*

Theorem 3. *[5] Exact cover is NP-complete.*

Theorem 4. *[4] Exact cover is equivalent to finding the 0-1 solution vector x of Ax=b where A is a 0-1 matrix and the vector b consists of only ones. The problem is NP-complete even when A is restricted to having two or three ones in each row.*

Note that the matrix multiplication above is defined over Z, not over Z_2. Hence in a solution of $Ax = b$ every equation has exactly one variable with value one and the rest have value zero.

We shall present a transformation from the restricted form of Exact cover (given i Theorem 4) to SCP. The structure of the reduction is similar to the one Johnson and Papadimitriou use when they reduce Exact cover to Hamiltonian cycle [4]. The Exact cover problem consists of several equations as in Theorem 4. The equations will be represented by a graph. If and only if the graph contains a spanning cactus the equations have a solution and the solution can be determined from the spanning cactus. We will do the reduction in three steps. First we will construct the corresponding graph, then show that if there is a solution to the equations we can find a spanning cactus in the graph, and thereafter prove that if there is a spanning cactus in the graph we can find a solution to the equations.

3.1.1 Construction of the corresponding graph

Each equation will be represented by a gadget in the graph. There are two types of equations and therefore two types of so called *equation-gadgets.* Every possible spanning cactus corresponds to a solution of the equations. Also each variable will be represented by a gadget. There are two possibilities for the spanning cactus in the *variable-gadget* which corresponds to the value of the variable. To ensure consistency of the solution there is a so called *xor-gadget* which connects the variable-gadget with the equation-gadget where the variable occur.

There are two types of equations, $x_1 + x_2 + x_3 = 1$ and $x_1 + x_2 = 1$. Equations with three variables are represented by gadgets as in Figure 1. Each variable corresponds to an edge in the gadget. If a *variable-edge* is in the spanning cactus the variable is zero, otherwise it is one.

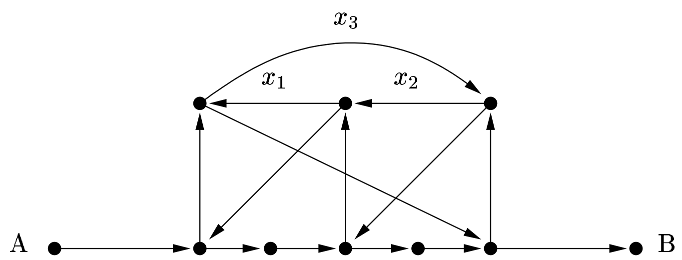

Figure 1: Gadget for equations with three variables $x_1 + x_2 + x_3 = 1$.

Equations with two variables are represented by gadgets as in Figure 2. Also here each variable corresponds to one edge in the gadget and if the variable-edge is in the spanning cactus the variable is zero otherwise it is one.

Each variable is represented by a gadget as in Figure 3. The value of the variable is represented by two edges. Only one of the *value-edges* can be in the spanning cactus (Lemma 6) and intuitively:

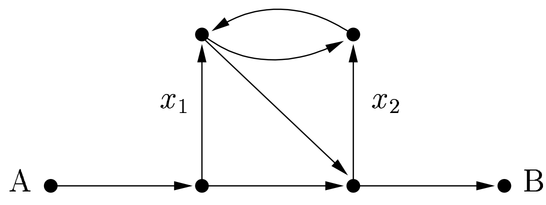

Figure 2: Gadget for equations with two variables $x_1 + x_2 = 1$

if the *zero-edge* is in the spanning cactus the variable has the value zero and if the *one-edge* is in the spanning cactus the variable has the value one.

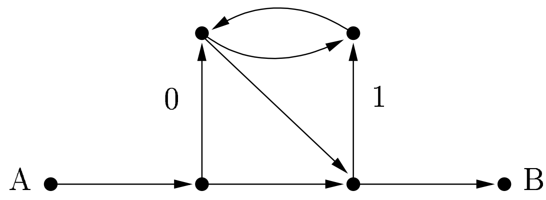

Figure 3: Variable-gadget.

All these gadgets are linked after eachother in a cycle (Figure 4).

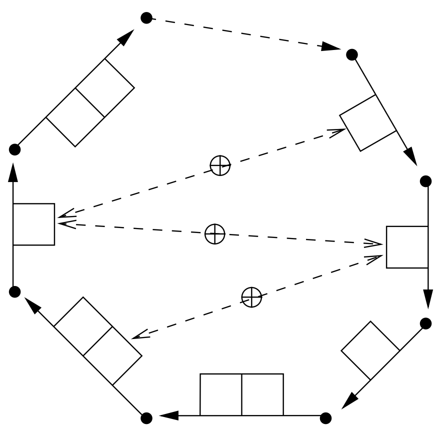

Figure 4: The structure of the graph. Equation-gadgets and variable-gadgets are linked in a cycle. The variable-edges in the equation-gadgets are connected to the one-edges in the variable-gadgets by xor-gadgets. (Most of the xor-gadgets are omitted in the figure.)

To ensure that a spanning cactus gives the same value to a variable in all equation-gadgets a variable-edge in an equation-gadget is connected to the one-edge in the variable-gadget by an xor-gadget as in Figure 5. The xor-gadget will have the property that exactly one of the two edges it connects can be in a spanning cactus (Lemma 7 and 8).

The inner structure of the xor-gadget is as in Figure 6. ABCD is the one-edge in the variable-gadget and LKJI is the variable-edge in the equation-gadget.

If one variable occurs in several equations the xor-gadgets are linked together in the variable-

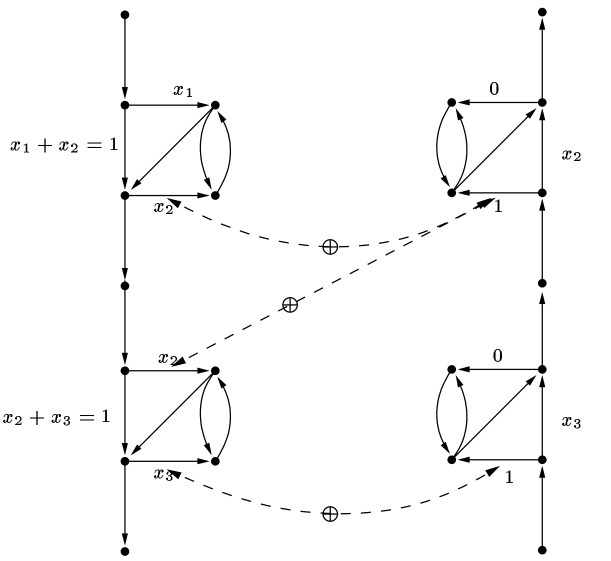

Figure 5: The xor connection between variable-edges in the equation-gadgets (left) and one-edges in the variable-gadgets (right).

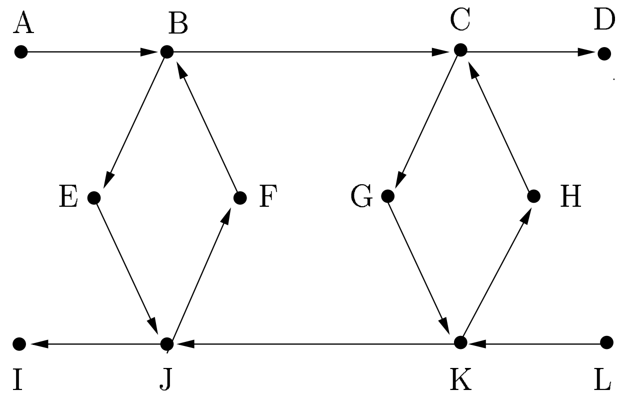

Figure 6: Xor-gadget. ABCD and LKJI are the edges which the xor-gadget connects.

gadget as in Figure 7. The figure shows two linked xor-gadgets but it can be extended to arbitrarily many. In Figure 7 AF is the one-edge in the variable-gadget, RO and VS are variable-edges in the equation-gadgets. In detail the linked xor-gadgets look like Figure 8.

3.1.2 A solution to Exact cover gives a spanning cactus

If there is a solution to Exact cover we want it to be a spanning cactus in our constructed graph. In this section we will prove this by showing how to find a spanning cactus from a solution of the equations in Exact cover.

Lemma 2. *If there is a solution to the Exact cover problem then there is a spanning cactus in the constructed graph.*

Proof. In a solution of an equation with three variables, two have the value zero and one has the

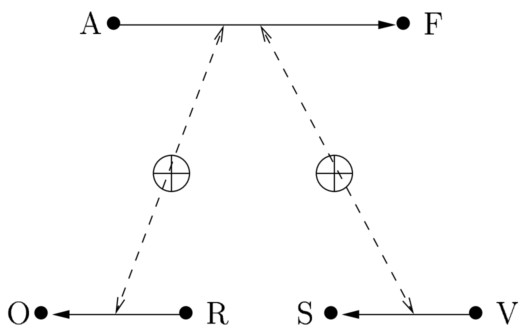

Figure 7: Linked xor-gadgets. If a variable occurs in two different equations the xor-gadgets are linked. AF is the one-edge in the variable-gadget, RO and VS are variable-edges in the equation-gadgets.

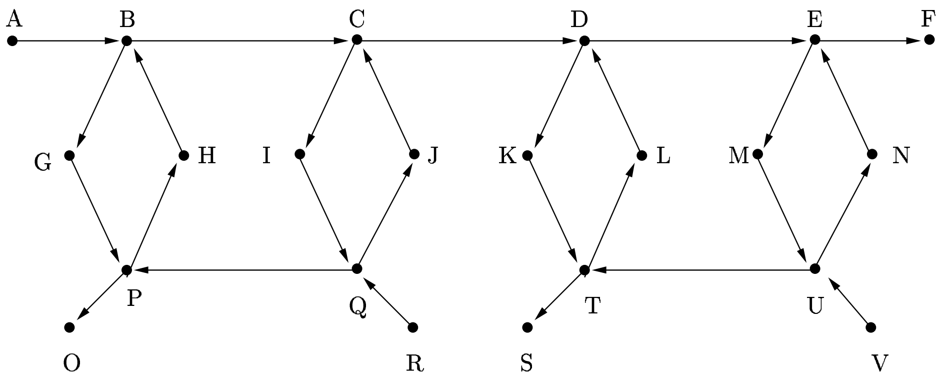

Figure 8: Linked xor-gadgets (the same as Figure 7)

value one. If a variable has the value zero its variable-edge is in the spanning cactus otherwise it is not. Figure 9 shows three cactus branches which include exactly two of the three variable-edges in the equation-gadget.

In the equations with two variables a solution has exactly one variable with the value one and one with the value zero. Figure 10 shows two cactus branches which include exactly one of the two variable-edges in the equation-gadget.

In a solution a variable obviously has the value zero or one. The corresponding edge in the variable-gadget (Figure 3) is in the cactus branch. The variable-gadget and the equation-gadget for two variables are identical and the two cactus branches in Figure 10 are the same for the variable-gadget. Thus we can find a cactus branch in the variable-gadget for each value of the variables.

In a solution a variable has a unique value and thus the xor-gadgets will only connect edges in the spanning cactus with edges not in the spanning cactus. Figure 11 shows cactus branches in the xor-gadget which includes exactly one of the two "edges" ABCD and LKJI.

Hence if there is a solution to the equations $Ax = b$ we can find a spanning cactus in the constructed graph. □

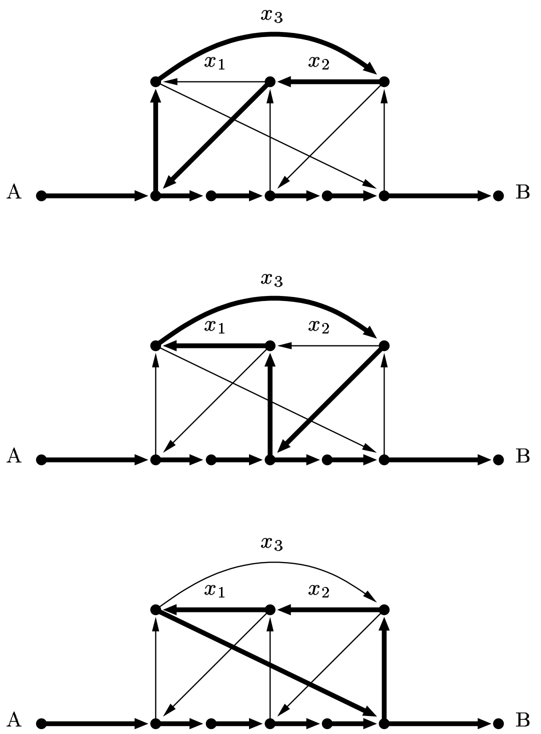

Figure 9: Possible cactus branches in a gadget for equations with three variables, $x_1 + x_2 + x_3 = 1$.

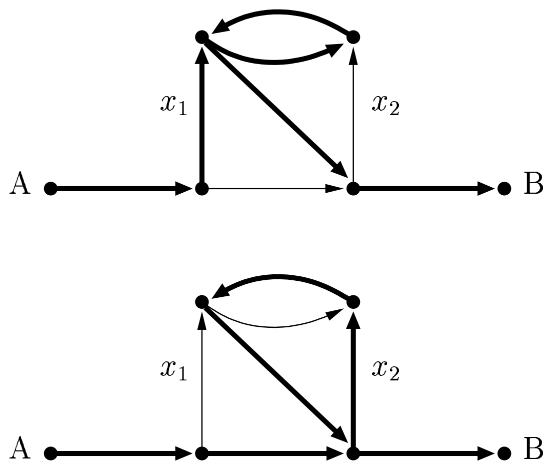

Figure 10: Possible cactus branches in a gadget for equations with two variables, $x_1 + x_2 = 1$.

3.1.3 A spanning cactus gives a solution to Exact cover

In this section we will prove that if there is a spanning cactus in our constructed graph there is a solution to the equations in Exact cover (Lemma 3). We will also show how to determine the solution from the spanning cactus.

Lemma 3. *Suppose there is a spanning cactus in our constructed graph. Then there is a solution to the equations $Ax = b$ and the solution can be found in the spanning cactus.*

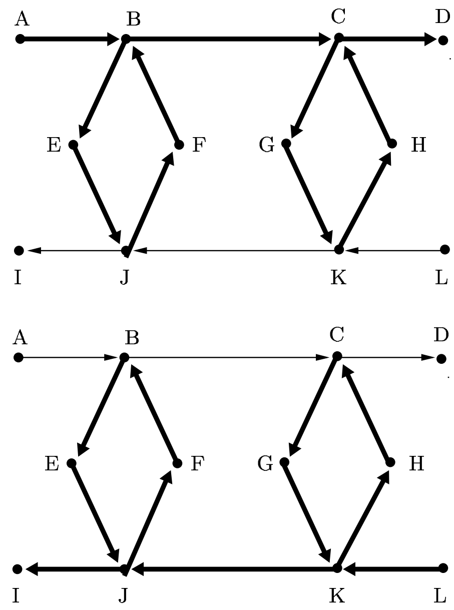

Figure 11: Cactus branches in a xor-gadget.

It is easy to see that the vertices connecting the variable- and equation-gadgets are in the spanning cactus (Figure 4) since it is strongly connected. Some edges in the variable- and equation-gadgets are not really edges but xor-gadgets. Presently we view them as atomic edges and prove in Lemma 7 and 8 that our view holds.

Suppose there is a spanning cactus in our constructed graph. Then there is a cactus branch in every gadget. We will prove that every cactus branch corresponds to a solution and that the solution of the equations is consistent. We need the following properties of a directed cactus:

Lemma 4. *In a directed cactus every vertex has the same in- and out-degree.*

Proof. Assume that for one vertex there are more out- than in-edges. Since the graph is strongly connected every out-edge is the beginning of a cycle. By the pigeon-hole principle at least two cycles ends at the same in-edge. Then the in-edge is contained in two cycles which contradicts the definition of a cactus. The same type of argument holds if there are more in- than out-edges. □

In the equation-gadget with three variables every cactus branch corresponds to a solution of the equation. In other words exactly two of the three variable-edges should be in the cactus branch. The following lemma proves this and that the cactus branches in Figure 9 are the only possible cactus branches in the gadget.

Lemma 5. *Suppose that the gadget for equations with three variables (Figure 1) is a subgraph in an arbitrary graph. Vertices A and B are connected to the rest of the graph but no other vertices have any other edges than the ones in the figure. Any cactus branch includes exactly two of the edges x_1, x_2 and x_3.*

Proof. The path is restricted in several ways. It follows the lower horizontal edges to connect all vertices. If it traverses one vertical edge the cycle has to end in the same vertex to make the in- and out-degree equal (Lemma 4). The spanning cactus traverses exactly one of the vertical edges (otherwise one edge is contained in more than one cycle). For each vertical line there is exactly one way to connect all vertices and to give all vertices an equal in- and out-degree (Figure 9). □

In the equation-gadget with two variables every cactus branch includes exactly one of the two variable-edges to correspond to a solution of the equation. The following lemma proves this and Figure 10 shows that the only possible cactus branches in the gadget.

Lemma 6. *Suppose an equation-gadget for equations with two variables (Figure 2) is a subgraph in an arbitrary graph. Vertices A and B are connected to the rest of the graph but no other vertices have any other edges than the ones in the figure. Any cactus branch includes exactly one of the edges x_1 and x_2.*

Proof. Since all vertices in a cactus have the same in- and out-degree (Lemma 4) there are only two possible ways to traverse the gadget (Figure 10). □

A variable should of course have exactly one value. In other words exactly one of the value-edges is in the cactus branch. The variable-gadget and the equation-gadget with two variables are identical and Lemma 6 ensures that any spanning cactus includes exactly one of the two value-edges.

We introduce the xor-gadget to ensure that the variable has the same value in all equations. Specially we want the xor-gadget to force exactly one of the edges it connects to be in the spanning cactus. The following lemma proves this and that the cactus branches in Figure 11 are the only ones.

Lemma 7. *Suppose that the xor-gadget (Figure 6) is a subgraph in an arbitrary graph. Vertices A, D, I and L are connected to the rest of the graph but no other vertices have any other edges than the one in the figure. Any cactus branch contains either the edges AB and CD but not JI and LK or it contains JI and LK but not AB and CD.*

Proof. Since the spanning cactus is strongly connected the two *diamonds* (BEJFB and CGKHC) are forced to be in the spanning cactus. A path in the cactus which starts in vertex A will end in vertex D (Figure 11) since every vertex in a cactus has the same in- and out-degree (Lemma 4). For the same reason a path which starts in vertex L ends in vertex I (Figure 11).

Assume that in the cactus there is a path from A to D and from L to I. Then particularly the edges BC and KJ are in the cactus (otherwise some vertices will not have the same number of in- and out-edges). Then the *diamonds* and the edges BC and KJ will form three different cycles. The edges CG, GK, JF and FB will be contained in two cycles which contradicts the definition of a cactus. □

If one variable occurs in several equations the xor-gadgets are linked together in the variable-gadget (Figure 8). Even for linked xor-gadgets Lemma 7 holds. More formally the Lemma can be extend to:

Lemma 8. *In an arbitrary graph two (or more) xor-gadgets linked as in Figure 8 form a subgraph. Single vertices as A, F, O, R, S and V (and possible more) are connected to the rest of the graph but no other vertices have any other edges than the ones in the figure. Any cactus branch contains either AB and EF or it contains RQ, PO, VU and TS (and possible more).*

Proof. All *diamonds* are in the spanning cactus since it is strongly connected. Since the in- and out-degree has to be equal (Lemma 4) a path in the cactus which starts in A still has to end in F, and the same goes for paths from R to O and from V to S (and possible more).

If there is a path in the cactus from A to F by Lemma 7 the edges RQ, PO, VU and TS can not be in the cactus.

If there is a path in the cactus from R to O we want to show that it has to be a path in the cactus from V to S as well. If there is a path from R to O the edges AB, BC, CD can not be in the spanning cactus by Lemma 7. If the edge CD is not in the cactus Lemma 7 shows that the edges VU and TS has to be in the cactus and that the edge EF can not be in the cactus. By induction the argument can be extended to arbitrary many xor-gadgets. □

To conclude: If there is a spanning cactus in the graph every variable-gadget gives a value to the corresponding variable (Lemma 6). The construction of the xor-gadgets ensures that every variable has the same value in all equations (Lemma 7). Since there is a spanning cactus every equation is satisfied (Lemma 5) and (Lemma 6). Thus we have a solution of $Ax = b$ and have proven Lemma 3.

3.2 Conclusion

We have constructed a graph from a the equations $Ax = b$ and shown that if there is a solution to the equations we can find a spanning cactus in the graph (Lemma 2). If there is a spanning cactus in the graph Lemma 3 shows that we can find a solution to the equations by the spanning cactus. The result can be formalized to:

Theorem 5. *There is a spanning cactus in the constructed graph if and only if $Ax = b$ has a solution.*

SCP is in NP (Lemma 1) and the reduction from Exact cover to SCP can obviously be done in polynomial time. Since Exact cover is known to be NP-complete [5], Theorem 5 proves that SCP also is NP-complete (Theorem 2) and we are done.

Acknowledgments

I thank Lars Engebretsen and Mikael Goldmann for great help and support in developing the ideas and writing the paper. Lena Folke and Simon Wigzell have helped me with my Achilles' heal: English — thank you!

References

[1] Nicos Christofides Worst case analysis of a new heuristic for the traveling salesman problem *Tech. Rep.* 388, GSIA, Carnegie Mellon Univerity, 1976.

[2] Lisa Fleischer Building Chain and Cactus Representations of All Minimum Cuts from Hao-Orlin in the Same Asymptotic Run Time. *Journal of Algorithms* 1:51-72, 1999.

[3] Alan M. Frieze, Giulia Galbiati and Francesco Maffioli On the Worst-Case Preformance of Some Algorithms for the Asymmetric Traveling Salesman Problem *Networks* 12:23-39, 1982.

[4] David S. Johnson, Christos H. Papadimitriou Computational complexity In Eugene L. Lawler, Jan Karel Lenstra, Alexander H. G. Rinnooy Kan and David B. Shmoys, editors, *The Traveling Salesman Problem : A Guided Tour of Combinatorial Optimization* 37-85, 1985. John Wiley & Sons. ISBN 0-471-90413-9.

[5] Richard M. Karp Reducibility among combinatorial problems In Raymond E Miller and James W Thatcher, editors, *Complexity of Computer Computations* 85-103, Plenum Press, New York, 1972.

[6] Christos H. Papadimitriou and Santosh Vempala On the approximability of the traveling salesman problem manuscript may 2002. `http://www.cs.berkeley.edu/~christos/tsp.ps`

[7] Sartaj K. Sahni and Teofilo Gonzalez P-complete approximation problems *Journal of the Assoc. Comput. Mach.* 23, 3, 555-556, 1976.

[8] Günter Schaar Remarks on Hamiltonian properties of powers of digraphs *Discrete Applied Mathematics* 51:181-186, 1994.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'