Discrete Applied Mathematics 156 (2008) 2573–2580

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Finding Hamiltonian circuits in quasi-adjoint graphs

Jacek Blazewicz [a,b], Marta Kasprzak [a,b], Benjamin Leroy-Beaulieu [c,*], Dominique de Werra [c]

[a] Institute of Computing Science, Poznan University of Technology, Poznan, Poland

[b] Institute of Bioorganic Chemistry, Polish Academy of Sciences, Poznan, Poland

[c] IMA-ROSE, Ecole Polytechnique Fédérale de Lausanne, Switzerland

ARTICLE INFO

Article history:
Received 30 November 2006
Received in revised form 16 July 2007
Accepted 6 March 2008
Available online 2 May 2008

Keywords:
Hamiltonian circuits
Quasi-adjoint graphs

ABSTRACT

This paper is motivated by a method used for DNA sequencing by hybridization presented in [Jacek Blazewicz, Marta Kasprzak, Computational complexity of isothermic DNA sequencing by hybridization, Discrete Appl. Math. 154 (5) (2006) 718–729]. This paper presents a class of digraphs: the quasi-adjoint graphs. This class includes the ones used in the paper cited above. A polynomial recognition algorithm in $O(n^3)$, as well as a polynomial algorithm in $O(n^2 + m^2)$ for finding a Hamiltonian circuit in these graphs are given. Furthermore, some results about related problems such as finding a Eulerian circuit while respecting some forbidden transitions (a path with three vertices) are discussed.

1. Introduction

DNA sequencing problems have been widely studied and, in particular, various formulations have been given in terms of combinatorial optimization of graph theoretical flavor (see references in [6]).

In [7], Blazewicz and Kasprzak develop a formulation involving the search of a Hamiltonian path in order to solve a problem of DNA sequencing. They exhibit cases where the problem can be solved in polynomial time. The graphs they use are a generalization of directed line graphs. It is interesting to examine how we can generalize those graphs, while still being able to solve the Hamiltonian Path or Circuit Problem polynomially. We give here a characterization of quasi-adjoint graphs, and devise a polynomial algorithm for finding a Hamiltonian circuit.

Graph theoretical terms not defined here can be found in [2].

2. Preliminaries

A *simple path* P in a graph $G = (V, U)$ is a sequence $\langle x_1, x_2, \ldots, x_k\rangle$ of distinct vertices from V such that $(x_i, x_{i+1}) \in U$ for $1 \le i \le k-1$. A *Hamiltonian path* in G is a simple path that includes all the vertices of V. A *Hamiltonian circuit* is a Hamiltonian path such that the edge (x_k, x_1) is in U.

The problem of deciding whether a graph has a Hamiltonian circuit (for short, the Hamiltonian Circuit Problem) has been known for a long time to be NP-complete [13]. In other words, the problem belongs to a large class of computationally related problems, for which no algorithm is known whose running time is bounded by a polynomial in the size of the input.

The Hamiltonian Circuit Problem remains NP-complete even for graphs having a specific structure, such as planar-cubic 3-connected graphs [11], bipartite planar graphs of maximum degree 3 [1], grid graphs [12], maximal planar graphs [8], chordal bipartite graphs and strongly chordal split-graphs [17] as well as line graphs [3].

* Corresponding author.
E-mail address: benjamin.leroy-beaulieu@epfl.ch (B. Leroy-Beaulieu).

0166-218X/$ – see front matter
doi:10.1016/j.dam.2008.03.014

However, for some other classes of graphs, such as locally connected regular graphs of degree 5 [15], cographs [9], proper circular arc graphs [4], interval graphs [14], co-comparability graphs [10] or directed line graphs [5], the same problem has been shown to be polynomially solvable.

3. Definitions and characterization

Throughout this paper, the symbol $\subset$ always refers to a *strict* inclusion. If an inclusion is not strict, we will use the symbol $\subseteq$.

Furthermore, in the remaining part of this paper, we will only consider directed graphs, even when not explicitly stated.

Definition 1. For any graph $G = (V, U)$, we define $n = |V|$ and $m = |U|$.

Definition 2. A *subpath* of a path P is a sequence of vertices which are consecutive in P.

Definition 3. A *transition* is a path consisting of 3 vertices.

Definition 4. Let $G = (V, U)$ be a graph and $x \in V$. Define $N^+(x)$ and $N^-(x)$ as follows: $N^+(x) = \{y \in V | (x, y) \in U\}$ and $N^-(x) = \{y \in V | (y, x) \in U\}$.

$N^+(x)$ is called the set of *successors* of x and $N^-(x)$ is called the set of *predecessors* of x.

For a set S of vertices, $N^+(S) = \cup_{x\in S} N^+(x)$.

Definition 5. Let $G = (V, U)$ be a graph and $x \in V$. We define the outdegree $d^+(x)$ (respectively the indegree $d^-(x)$) of a vertex x as the number of arcs leaving (respectively entering) x. Formally:

$$d^+(x) = |\{u \in U | u = (x, y) \text{ for some } y \text{ of } V\}|$$
$$d^-(x) = |\{u \in U | u = (y, x) \text{ for some } y \text{ of } V\}|.$$

Remark 1. Since graphs considered in this paper may be multigraphs, $d^+(x)$ may be different from $|N^+(x)|$ and $d^-(x)$ may be different from $|N^-(x)|$.

Definition 6. A graph is a *quasi-adjoint graph* if the family $(N^+(y) | y \in V)$ is nested. In other words, if for any two vertices x and y the following property holds:

$$N^+(x) \cap N^+(y) \neq \emptyset \Rightarrow \begin{array}{l} N^+(x) = N^+(y) \text{ or} \\ N^+(x) \subset N^+(y) \text{ or} \\ N^+(y) \subset N^+(x). \end{array}$$

Remark 2. Berge [2] gives the following definitions:

A graph is a p-graph if given any ordered pair x, y of vertices (x possibly equal to y), there are at most p parallel arcs from x to y.

The *adjoint* $G = (V, U)$ of a graph $H = (X, V)$ is the 1-digraph with vertex set V and such that there is an arc from a vertex x to a vertex y in G if and only if the terminal endpoint of arc x in H is the initial endpoint of arc y in H.

A graph G is an *adjoint* if there exists some graph H such that G is the adjoint of H.

Berge [2] also proves that a 1-graph $G = (V, U)$ is the adjoint of a graph if and only if the following holds for any pair x, y of vertices in V:

$$N^+(x) \cap N^+(y) \neq \emptyset \Rightarrow N^+(x) = N^+(y).$$

Remark 3. This statement shows that, by definition, the class of quasi-adjoint graphs strictly contains the class of adjoint graphs.

Remark 4. Quasi-adjoint graphs can be recognized in $O(n^3)$ time by looking at every pair of vertices x and y and comparing $N^+(x)$ and $N^+(y)$.

The following constructions and definitions will be used for the search of a Hamiltonian circuit in a quasi-adjoint graph G:

One can construct a new graph G' by splitting each vertex x of G into two new vertices x' and x'', and replacing each arc (x, y) by the arc (x'', y'). An example of this is given in Fig. 1.

Definition 7. Each non-trivial connected component (having at least two vertices) of G' is called a *cluster*.

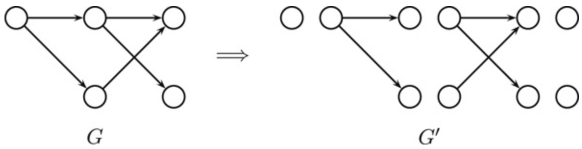

Fig. 1. A quasi-adjoint graph G and the result of its decomposition G'. This figure also illustrates Remark 3: while adjoint graphs would only admit complete-bipartite graphs after a decomposition, quasi-adjoint graphs also admit some incomplete-bipartite graphs.

Remark 5. G' is a collection of vertex-disjoint bipartite graphs and isolated vertices. By definition, the clusters are the bipartite graphs.

For each cluster C, we divide its set of vertices into two parts: the *left part* $L(C)$ is the set of vertices having only outgoing arcs and the *right part* $R(C)$ is the set of vertices having only incoming arcs.

Note that the clusters resulting from the decomposition of a quasi-adjoint graph are not necessarily complete, as they would be for the adjoint of a graph. See Fig. 1 for an example. It is possible to group vertices of $L(C)$ into subsets such that, for any two vertices x and y from the same subset, $N^+(x) = N^+(y)$. As a direct consequence of the definition of quasi-adjoint graphs (Definition 6), each one of these subsets then belongs to one of the following categories:

$\mathcal{A}$. $\{x|N^+(x) = R(C)\}$
$\mathcal{B}$. $\{x|\exists y, z : N^+(y) \subset N^+(x) \subset N^+(z)\}$ $(x, y, z \in L(C))$
$\mathcal{C}$. $\{x|\exists z : N^+(x) \subset N^+(z)$ and $\nexists y : N^+(y) \subset N^+(x)\}$ $(x, y, z \in L(C))$.

Lemma 1. *For every cluster C, there is at least one vertex $x \in L(C)$ such that $N^+(x) = R(C)$.*

Proof. Suppose there exists a cluster C such that there exists no vertex $x \in L(C)$ with $N^+(x) = R(C)$. Consider two disjoint maximal sets Y_1 and Y_2 such that $Y_i = N^+(x_i)$ for some $x_i \in L(C)$. Since the family $\cup_i Y_i$ is nested, there is no chain going from x_1 to x_2. Thus, the cluster is disjoint, which is a contradiction. □

4. The Hamiltonian circuit problem in quasi-adjoint graphs

Theorem 1. *The Hamiltonian Circuit Problem in quasi-adjoint graphs can be polynomially solved in $O(n^2 + m^2)$ time.*

Proof. We prove this by giving the Algorithm 1, which finds a Hamiltonian circuit in a quasi-adjoint graph if there is one and gives a negative answer otherwise.

This algorithm is based on the same construction as the one used for adjoint graphs: transforming graph G into its original graph H (such that G is the adjoint of H) and then looking for a Eulerian circuit in H. However, since clusters of quasi-adjoint graphs are not necessarily complete, as shown in Fig. 1, we must introduce some artificial vertices to H to make sure that the one-to-one correspondence between a Eulerian circuit in H and a Hamiltonian circuit in G remains. In other words, do not make H Eulerian if G was not Hamiltonian, nor make H not Eulerian if G was Hamiltonian. Algorithm 1 is illustrated in Figs. 2 and 3.

Remark 6. In Algorithm 1, at step 26, each labeled arc in H corresponds to a vertex of the same name in G.

Claim 1. *Step* 8 *of Algorithm 1 constructs a directed tree T with root Y_1 and the sets of type $\mathcal{C}$ as leaves.*

Proof. The arcs of T represent a relation of inclusion: an arc from Y_i to Y_j implies that $Y_j \subset Y_i$. Furthermore, if $Y_j \subset Y_i$, then there is a path from Y_i to Y_j in T. Since Y_1 contains all other sets Y_i, T is connected and Y_1 is a root of T. Since the sets of type $\mathcal{C}$ contain no other subsets, they are leaves of T. Finally, since an arc of T is a relation of strict inclusion, in the sense that an arc from Y_i to Y_j implies that there is no set Y_k such that $Y_j \subset Y_k \subset Y_i$. Thus, if $Y_j \subset Y_i$, then there is only one path from Y_i to Y_j in T and therefore T is a tree. □

Claim 2. *At Steps* 5 *and* 18*, Algorithm 1 exits only if there is no Hamiltonian circuit in G.*

Proof. Suppose there is a Hamiltonian circuit in G. By construction of the cluster C, the edges that belong to both the Hamiltonian circuit of G and the cluster C define a perfect matching in C. If $|L(C)| \neq |R(C)|$, C does not admit a perfect matching and therefore, G does not admit a Hamiltonian circuit. This ends the proof of Claim 2 for step 5.

Algorithm 1 builds the vertices k_j such that there is a path from every vertex $x'' \in X_i$ such that $Y_i \supset Y_j$ to k_j and there is a path from k_j to every vertex $y' \in Y_j$ (including the subsets of Y_i).

At step 16, all arcs exiting k_j have been built, and at least $|X_j|$ must enter k_j. If $|X_j| > d^+(k_j)$, it means that

$$|\cup_{k|Y_k \subseteq Y_j} X_k| > |Y_j|.$$

Since the vertices in $\cup_{k|Y_k \subseteq Y_j} X_k$ do not have any other successor but the vertices in Y_j, this means that there is no possible Hamiltonian circuit in G. □

Algorithm 1

Input: A quasi-adjoint graph $G = (V, U)$
Output: A Hamiltonian circuit in G or a claim of non-existence of such a circuit
1: Define an empty graph $H = (V^H, U^H)$ with $V^H \leftarrow \emptyset$ and $U^H \leftarrow \emptyset$
2: For every $x \in V$, introduce two vertices x' and x'' into V^H. For every arc $(x, y) \in U$ such that $x \neq y$, introduce the arc (x'', y') in U^H.
3: **for** each cluster C of H, **do**
4: **if** $|L(C)| \neq |R(C)|$ **then**
5: Exit. There is no Hamiltonian circuit in G.
6: **end if**
7: Decompose $L(C)$ into sets of types $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$. The unique set of type $\mathcal{A}$ is labeled X_1. Label all other sets of $L(C)$ with a unique identifier X_i. Let $Y_i \leftarrow N^+(X_i)$.
8: Sort sets Y_i according to their inclusion relation: Construct a directed tree $T = (V^T, U^T)$ with $V^T = \cup_i\{Y_i\}$ and $(Y_i, Y_j) \in U^T \Leftrightarrow Y_j \subset Y_i \bigwedge \nexists k : Y_j \subset Y_k \subset Y_i$.
9: Delete all arcs of C.
10: For every leaf Y_i of T, introduce a vertex k_i in V^H. For each vertex $x'' \in X_i$, add the arc (x'', k_i) to U^H and for each vertex $y' \in Y_i$, add the arc (k_i, y') to U^H. Label the vertex Y_i of T.
11: **for** each vertex Y_i of T not yet labeled such that all elements of $N^+(Y_i)$ in T are labeled **do**
12: Introduce a vertex k_i into V^H.
13: For each vertex $x'' \in X_i$, introduce the arc (x'', k_i) into U^H.
14: For each vertex $y' \in Y_i$ such that $N^-(y') = \emptyset$, introduce the arc (k_i, y') into U^H.
15: **for** each $Y_j \in N^+(Y_i)$ in T **do**
16: $o_j \leftarrow d^+(k_j) - |X_j|$.
17: **if** $o_j < 0$ **then**
18: Exit. There is no Hamiltonian circuit in G.
19: **else**
20: add o_j arcs (k_i, k_j) to U^H.
21: **end if**
22: **end for**
23: Label the vertex Y_i of T.
24: **end for**
25: **end for**
26: In H, link each pair of vertices x' and x'' by the arc (x', x'') and label this arc x.
27: Search for a Eulerian circuit in H. If there is none, there is no Hamiltonian circuit in G. Otherwise, the (closed) sequence of labels of arcs from the Eulerian circuit is the solution for the Hamiltonian Circuit Problem in G.

Claim 2 shows that the Algorithm exits before reaching step 27 only if it could be shown before that there is no Hamiltonian circuit in G. We now have to show that step 27 finds a Eulerian circuit in H if and only if there is a Hamiltonian circuit in G. in order to do so, we start with proving Claims 3 and 4.

Claim 3. *For any two vertices x'' and y' of the same cluster, there is a path from x'' to y' in H if and only if there is an arc (x, y) in G.*

Proof. $\Rightarrow$ Suppose that there is an arc (x, y) in G. Let X_i be the set containing x''. Then, by construction, $Y_i \ni y'$.

If X_i is of type $\mathcal{C}$, then Algorithm 1 builds a path $< x'', k_i, y' >$ at step 10.

Consider now the case where X_i is of type $\mathcal{A}$ or $\mathcal{B}$. If $\nexists j : y' \in Y_j \bigwedge Y_j \subset Y_i$, then, at the time when Y_i is chosen at step 11, no Y_j containing y' has been chosen before and thus no arc entering y' has been added to U^H yet; thus $N^-(y') = \emptyset$; so the algorithm builds a path $< x'', k_i, y' >$ at step 14.

Finally, if $\exists j : y' \in Y_j \bigwedge Y_j \subset Y_i$, then there are the arcs (x'', k_i) and (k_j, y') in U^H. Note that, at step 20, the algorithm puts an arc (k_l, k_m) in U^H if there is an arc (Y_l, Y_m) in T. Thus, since $Y_j \subset Y_i$, there is a path from Y_i to Y_j in T and also a path from k_i to k_j in H. Thus, there is a path from x'' to y' in H.

$\Leftarrow$ Suppose that there is a path from x'' to y' in H. By construction, x'' has only one successor, that we denote k_i and y' has only one predecessor, that we denote k_j. This means that $x'' \in X_i$ and $y' \in Y_j$. If $i = j$, then $Y_j = Y_i = N^+(X_i)$. Thus, there is an arc (x, y) in G. Else, it means that there is a path from k_i to k_j. Thus, there is a path from Y_i to Y_j in T and thus, $Y_j \subset Y_i$ and $y \in Y_i$, which implies that (x, y) is in G. $\square$

Claim 4. *At the end of Algorithm 1, every vertex $v \in V^H$ that was part of some cluster at the end of step 2, as well as every vertex that was introduced by the algorithm (the k_i vertices), satisfies $d^+(v) = d^-(v)$.*

Proof. There are three types of such vertices: the vertices x' and x'', and the vertices k_i, for every i.

For every vertex x', there is only one incoming arc, added at step 10 or 14. There is also only one outgoing arc, which is (x', x'') (step 26). Thus, $d^+(x') = d^-(x')$.

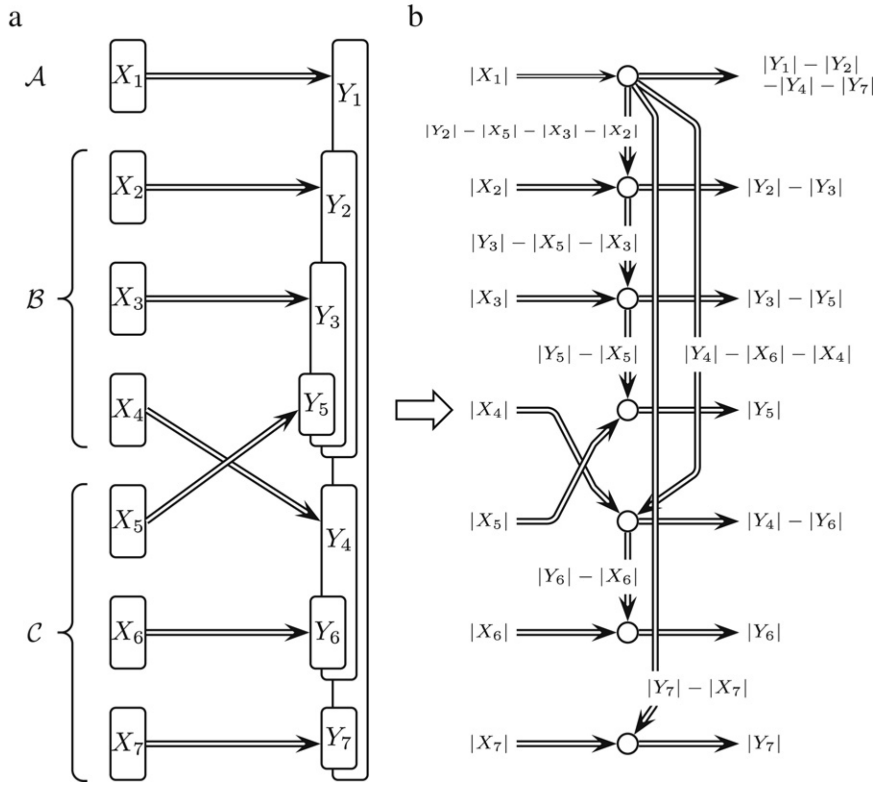

Fig. 2. Illustration of the transformation of a cluster by Algorithm 1 (Steps 3 to 25). Part a represents the original cluster coming from the decomposition of graph G and part b represents the resulting part of H. The numbers next to the arcs represent the multiplicity of these arcs.

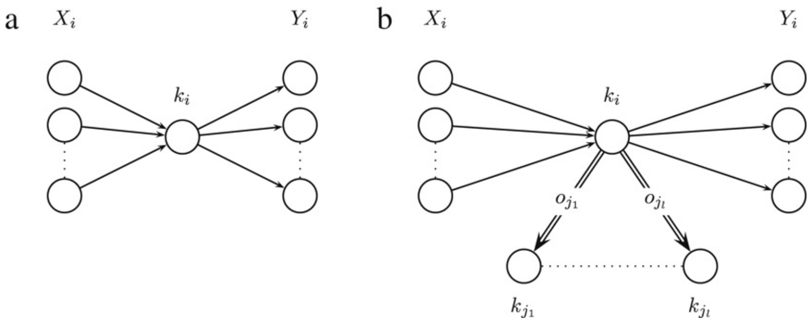

Fig. 3. Construction by Algorithm 1. Part a: step 10 (type $\mathcal{C}$). Part b: steps 12 through 22 (types $\mathcal{A}$ and $\mathcal{B}$).

Similarly, for every vertex x'', there is only one incoming arc, which is (x', x'') (step 26). There is also only one outgoing arc, added at step 10 or at step 13. Thus, $d^+(x'') = d^-(x'')$.

Consider now a vertex k_i. We want to prove that $d^-(k_i) = d^+(k_i)$ for all i. For every i, we have:

$$d^+(k_i) = \underbrace{|Y_i| - \sum_{j|Y_j \in N^+(Y_i) \text{ in } T} |Y_j|}_{\text{by step 10 or 14}} + \underbrace{\sum_{j|Y_j \in N^+(Y_i) \text{ in } T} o_j}_{\text{by step 10 or 20}} . \tag{1}$$

For every $i \neq 1$, we have:

$$d^-(k_i) = \underbrace{|X_i|}_{\text{by step 13}} + \underbrace{o_i}_{\text{by step 20}} . \tag{2}$$

And for $i = 1$, we have:

$$d^-(k_1) = |X_1|. \tag{3}$$

Besides that, by step 16:

$$o_i = d^+(k_i) - |X_i| = |Y_i| - \sum_{j|Y_j \in N^+(Y_i) \text{ in } T} |Y_j| + \sum_{j|Y_j \in N^+(Y_i) \text{ in } T} o_j - |X_i|. \tag{4}$$

Let us now prove that, for all i,

$$o_i = |Y_i| - \sum_{j|\exists (Y_i \rightsquigarrow Y_j) \text{ in } T} |X_j| \tag{5}$$

where $(Y_i \rightsquigarrow Y_j)$ represents a path from Y_i to Y_j (possibly a path of length 0 if $i = j$). This is obvious if Y_i is a leaf of T. By recurrence, suppose that it holds for every Y_j such that $Y_j \in N^+(Y_i)$ in T. Then, Eq. (4) can be rewritten:

$$o_i = |Y_i| - \sum_{j|Y_j \in N^+(Y_i) \text{ in } T} |Y_j| + \sum_{j|Y_j \in N^+(Y_i) \text{ in } T} \left(|Y_j| - \sum_{k|\exists (Y_j \rightsquigarrow Y_k) \text{ in } T} |X_k| \right) - |X_i|$$

$$o_i = |Y_i| - \sum_{j|Y_j \in N^+(Y_i) \text{ in } T} \left(\sum_{k|\exists (Y_j \rightsquigarrow Y_k) \text{ in } T} |X_k| \right) - |X_i|$$

$$o_i = |Y_i| - \sum_{j|\exists (Y_i \rightsquigarrow Y_j) \text{ in } T} |X_j|.$$

Thus, for every i, we have:

$$d^+(k_i) = |Y_i| - \sum_{j|Y_j \in N^+(Y_i) \text{ in } T} |Y_j| + \sum_{j|Y_j \in N^+(Y_i) \text{ in } T} \left(|Y_j| - \sum_{k|\exists (Y_j \rightsquigarrow Y_k) \text{ in } T} |X_k| \right)$$

$$d^+(k_i) = |Y_i| - \sum_{j|\exists (Y_i \rightsquigarrow Y_j) \text{ in } T} |X_j| + |X_i| \tag{6}$$

and, for every $i \neq 1$, by replacing o_i by its value in Eq. (2):

$$d^-(k_i) = |X_i| + |Y_i| - \sum_{j|\exists (Y_i \rightsquigarrow Y_j) \text{ in } T} |X_j| \tag{7}$$

which, with (6), leads to the conclusion that, for $i \neq 1$, $d^+(k_i) = d^-(k_i)$.

For $i = 1$, $d^-(k_1) = |X_1|$. Besides that, step 4 ensures that

$$|Y_1| = \sum_{j|\exists (Y_1 \rightsquigarrow Y_j) \text{ in } T} |X_j|. \tag{8}$$

Thus, Eq. (6) becomes $d^+(k_1) = |X_1|$ and, by (3), $d^+(k_1) = d^-(k_1)$. This ends the proof of Claim 4. □

Remark 7. The vertices x' (respectively x'') which were not part of any cluster at the end of step 2 have $d^-(x') = 0$ and $d^+(x') = 1$ (respectively $d^-(x'') = 1$ and $d^+(x'') = 0$). Of course, if any such vertex exists, G does not contain a Hamiltonian circuit.

Claim 5. *Algorithm 1 has a complexity of* $O(n^2 + m^2)$.

Proof. Step 1 has complexity 1. Step 2 has complexity $n+m$. Step 26 has complexity n and step 27 can be done in $O(n^2)$ [16].

For steps 3 to 25, since they are executed on disjoint parts of the graph, we will caculate their execution time over all passes through this loop instead of for each pass through this loop seperately.

Steps 4 to 6 have complexity $2n$. Steps 7 and 8 can be done in m^2 for the creation of sets X_i and m^2 again for the comparison of sets Y_i and the construction of T (both can be done at the same time); thus the complexity of these two steps is $O(m^2)$. The complexity of step 9 is m. The complexity of step 10 is smaller or equal to $3n$.

The loop at step 11 will be executed at most n times. The complexity of step 12 is 1. The complexity of steps 13 and 14 is at most n each. The loop at step 15 is executed at most n times. The steps within this loop have complexity of $O(1)$ except the step 20, which may add at most n arcs over all passes since $\sum_j (d^+(k_j)) = |\{y : \exists j \text{ such that } y \in Y_j\}| \leq n$. Therefore, the complexity of the loop 11 is $O(n^2)$.

This gives us an overall complexity of $O(n^2 + m^2)$. □

Remark 8. In the special case where G is a 1-graph, we have that $m \leq n^2$, thus the complexity of Algorithm 1 is $O(n^4)$.

Claims 4 and 3 prove that there is a Eulerian circuit in H if and only if there is a Hamiltonian circuit in G. Since, by Claim 5, Algorithm 1 is polynomial, this ends the proof of Theorem 1. □

5. Generalizations of quasi-adjoint graphs

Quasi-adjoint graphs are interesting because of the polynomiality of finding a Hamiltonian circuit. This section discusses two related problems.

Algorithm 2, if used before Algorithm 1, enlarges the class of graphs for which the Hamiltonian Circuit Problem is polynomially solvable, since it can transform some graphs into quasi-adjoint graphs.

Theorem 2 gives an interpretation of Algorithm 1 in terms of forbidden transitions and shows a limitation to the generalization of this idea.

5.1. Removal of arcs

When searching for a Hamiltonian circuit in a graph, some arcs can safely be removed. We devise here the algorithm doing this and show that it does not affect the hamiltonicity of a graph.

Algorithm 2

Input: A graph $G = (V, U)$
Output: G with some arcs removed without changing its hamiltonicity.
Remove all loops (arcs of type (x, x)).
Split G into clusters. Denote the new graph $G' = (V', U')$.
for each cluster C of G' **do**
  Solve the problem of perfect matching in C.
  if a perfect matching is found **then**
    label all arcs composing the solution as N (Necessary).
    for every not-labeled arc (x, y) of C **do**
      Consider the subgraph induced by the removal of x and y from C.
      Solve the problem of perfect matching in it.
      if there is no solution to this problem **then**
        Remove the arc (x, y) from U' /**Thus, also from* C.*/ and the corresponding arc from U.
      else
        label all the arcs of this solution and the arc (x, y) as N.
      end if
    end for
  end if
end for

Claim 6. *Algorithm 2 removes an arc (x, y) from a graph G only if this arc cannot be part of any Hamiltonian circuit in G.*

Proof. For each cluster C of G' and $x'' \in L(C)$ and $y' \in R(C)$, $R(C)$ includes $N^+(x'')$ and $L(C)$ includes $N^-(y')$. Consider now that (x'', y') does not belong to any perfect matching in C. Then, if (x, y) is in some Hamiltonian circuit Γ in G, there exists a vertex $v \in G$ with $v'' \in L(C)$ which does not have a successor in Γ. This is a contradiction with the definition of a Hamiltonian circuit. Thus, (x, y) may not be part of any Hamiltonian circuit in G. □

5.2. About forbidden transitions

Definition 8. Searching for a path with *forbidden transitions* is searching for a path which does not contain a forbidden transition as a subpath.

The method of Blazewicz and Kasprzak [7] searches for a Eulerian path in polynomial time in graphs where some transitions are forbidden. Unfortunately, Theorem 2 states that this cannot be generalized.

Theorem 2. *Given any Eulerian graph H with a collection $\mathcal{F}$ of forbidden transitions, it is NP-complete to find a Eulerian path (or circuit) which does not contain any transition from $\mathcal{F}$.*

Proof. Consider the Hamiltonian Path (Circuit) Problem in a directed graph $G = (V, U)$. It is always possible to introduce arcs into G so that it becomes the adjoint of some Eulerian graph H. This can be done in polynomial time by comparing the successors of every pair of vertices x and y and adding missing arcs to have $N^+(x) \cap N^+(y) \neq 0 \Rightarrow N^+(x) = N^+(y)$. The number of arcs introduced is smaller than n^2. When transforming G into its original graph H, each arc that was added to G results in a transition (a path of three vertices, see Definition 3) in H. These transitions are labeled as forbidden. Then, there exists a Eulerian path (circuit) in H that does not contain any forbidden transitions if and only if G has a Hamiltonian path (circuit). □

Remark 9. Every cluster of graph G constructed at the beginning of Algorithm 1 can be seen as a complete bipartite graph with some missing arcs (see Fig. 1). In order to find a Hamiltonian circuit in G, one could add all those arcs to G. G would then be an adjoint, which could be transformed into its original graph in order to find a Eulerian circuit in it. However, there would be some forbidden transitions, corresponding to the newly added arcs. As stated in Theorem 2, this is in general difficult. The construction of Algorithm 1, though only applicable to quasi-adjoint graphs, avoids this problem.

6. Conclusion

We have defined the polynomial-time recognizable class of quasi-adjoint graphs, which extends the set of known graph classes for which the Hamiltonian Circuit Problem is polynomially solvable. We have provided a polynomial-time algorithm of complexity $O(n^2+m^2)$ solving the problem in these graphs, as well as another algorithm which provides some extension of this class with respect to the polynomial solvability of the Hamiltonian Circuit Problem. The class of quasi-adjoint graphs is a generalization of two known classes: the adjoints [2] and the graphs modeling the problem of isothermic DNA sequencing by hybridization [7].

Acknowledgements

The research has been partially supported by a grant from the Ministry of Science of Poland.

References

[1] Takanori Akiyama, Takao Nishizeki, Nobuji Saito, NP-completeness of the Hamiltonian cycle problem for bipartite graphs, J. Inf. Process. 3 (2) (1980) 73–76.
[2] Claude Berge, Graphs and Hypergraphs, North Holland, Amsterdam, 1976.
[3] Alan A. Bertossi, The edge Hamiltonian path problem is NP-complete, Inform. Process. Lett. 13 (4–5) (1981) 157–159.
[4] Alan A. Bertossi, Finding Hamiltonian circuits in proper interval graphs, Inform. Process. Lett. 17 (2) (1983) 97–101.
[5] Jacek Blazewicz, Alain Hertz, Daniel Kobler, Dominique de Werra, On some properties of DNA graphs, Discrete Appl. Math. 98 (1–2) (1999) 1–19.
[6] Jacek Blazewicz, Marta Kasprzak, Complexity of DNA sequencing by hybridization, Theoret. Comput. Sci. 290 (3) (2003) 1459–1473.
[7] Jacek Blazewicz, Marta Kasprzak, Computational complexity of isothermic DNA sequencing by hybridization, Discrete Appl. Math. 154 (5) (2006) 718–729.
[8] V. Chvátal, Hamiltonian cycles, in: The Traveling Salesman Problem, in: Wiley-Intersci. Ser. Discrete Math, Wiley, Chichester, 1985, pp. 403–429.
[9] D.G. Corneil, H. Lerchs, L. Stewart Burlingham, Complement reducible graphs, Discrete Appl. Math. 3 (1981) 163–174.
[10] Jitender S. Deogun, George Steiner, Polynomial algorithms for Hamiltonian cycle in cocomparability graphs, SIAM J. Comput. 23 (3) (1994) 520–552.
[11] M.R. Garey, D.S. Johnson, R. Endre Tarjan, The planar Hamiltonian circuit problem is NP-complete, SIAM J. Comput. 5 (4) (1976) 704–714.
[12] Alon Itai, Christos H. Papadimitriou, Jayme Luiz Szwarcfiter, Hamilton paths in grid graphs, SIAM J. Comput. 11 (4) (1982) 676–686.
[13] Richard M. Karp, Reducibility among combinatorial problems, in: Complexity of Computer Computations (Proc. Sympos., IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y, 1972), Plenum, New York, 1972, pp. 85–103.
[14] J. Mark Keil, Finding Hamiltonian circuits in interval graphs, Inform. Process. Lett. 20 (4) (1985) 201–206.
[15] P.B. Kikust, The Existence of a Hamiltonian Cycle in a Regular Graph of Degree 5, in: Latvian Mathematical Yearbook, vol. 16, 1975, pp. 33–38 (in Russian), 271, Izdat. Zinatne, Riga.
[16] Eugene L. Lawler, Combinatorial Optimization: Networks and Matroids, Holt, Rinehart and Winston, New York, 1976.
[17] Haiko Müller, Hamiltonian circuits in chordal bipartite graphs, Discrete Math. 156 (1–3) (1996) 291–298.