Open Journal of Discrete Mathematics, 2016, 6, 13-24
Published Online January 2016 in SciRes. http://www.scirp.org/journal/ojdm
http://dx.doi.org/10.4236/ojdm.2016.61003

Solving the Maximum Matching Problem on Bipartite $Star_{123}$-Free Graphs in Linear Time

Ruzayn Quaddoura

Department of Computer Science, Faculty of Information Technology, Zarqa University, Zarqa, Jordan
Email: ruzayn@zu.edu.jo

Received 6 April 2015; accepted 23 January 2016; published 26 January 2016

Abstract

The bipartite $Star_{123}$-free graphs were introduced by V. Lozin in [1] to generalize some already known classes of bipartite graphs. In this paper, we extend to bipartite $Star_{123}$-free graphs a linear time algorithm of J. L. Fouquet, V. Giakoumakis and J. M. Vanherpe for finding a maximum matching in bipartite $Star_{123}$, P_7-free graphs presented in [2]. Our algorithm is a solution of Lozin's conjecture.

Keywords

Bipartite Graphs, Decomposition of Graphs, Design and Analysis of Algorithms, Matching

1. Introduction

A matching M of a graph $G=(V,E)$ is a subset of edges with the property that no two edges of M share a common vertex. A matching is called induced if the subgraph of G induced by M consists of exactly M itself. The maximum matching problem is to find a matching with the maximum cardinality. Graph matching is one of the fundamental problems in combinatorial optimization because of its use in various fields such as computational biology [3], pattern recognition [4], computer vision [5], music information retrieval [6], and computational music theory [7]. For arbitrary graphs, it is known that this problem can be solved in $O(m\sqrt{n})$ time [8]. Moitra and Johnson gave an $O(n\log n)$ time algorithm on interval graphs [9]. In addition Alt, Blum, Mehlhorn, and Paul gave an $O(n^{1.5}\sqrt{m\log n})$ time algorithm on bipartite graphs [10]. In [11] Yu and Yang exhibited an $O(n)$ time algorithm for the maximum matching problem on cographs. This result was extended in [12] by Fouquet, Parfenoff and Thuillier to a wider class, namely the P_4-tidy graphs. Also the technique developed in [11] was used by Fouquet, Giakoumakis and Vanherpe in [2] to find an $O(n)$ time algorithm for the maximum matching problem on bipartite $P_7, Star_{123}$-free graphs (see **Figure 1**). In [1], Lozin studied the class of bipartite

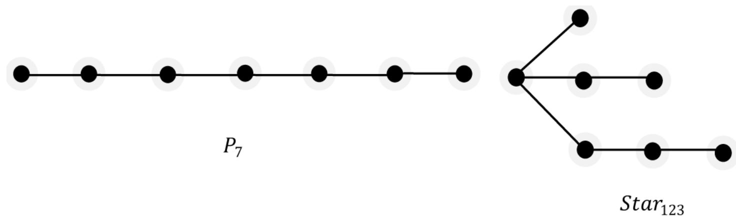

Figure 1. Forbidden configurations for $P_7, Star_{123}$-free graphs.

$Star_{123}$-free graphs and conjectured that both maximum induced matching problem and maximum matching problem in this class can be solved in linear time. The first one has been solved in [13]. In this paper we shall solve Lozin's conjecture for maximum matching problem by extending the matching algorithm for the class of $P_7, Star_{123}$-free graphs [2] to the class of bipartite $Star_{123}$-free graphs. Our algorithm is based on the recognition algorithm of the class $Star_{123}$-free bipartite graphs introduced by Quaddoura in [14].

2. Definitions and Properties

For terms not defined in the paper the reader can refer to [15]. The graphs considered in this paper are finite without multiple edges and loops. As usual, for any graph *G* we denote the set of its vertices by $V(G)$ and by $E(G)$ the set of its edges (or simply by *V* and *E* if there is no risk of confusion) and their respective cardinalities by *n* and *m*. A bipartite graph $G=(B\cup W,E)$ is defined by two disjoint vertex subsets *B* the black vertices and *W* the white ones, and a set of edges $E\subseteq B\times W$. The *bi-complement* of a bipartite graph $G=(B\cup W,E)$ is the bipartite graph defined by $\bar{G}^{bip}=(B\cup W,B\times W-E)$. If the color classes *B* and *W* are both non empty, the graph will be called *bichromatic*, *monochromatic* otherwise. A vertex *x* will be called *isolated* (resp. *universal*) if *x* has no neighbors in *G* (resp. in $\bar{G}^{bip}$). A *complete* bipartite graph is a graph having only universal white vertices and universal black vertices. A *stable* set is a subset of pairwise non-adjacent vertices. A chordless path on *k* vertices is denoted by P_k and a chordless cycle on *k* vertices is denoted by C_k. Given a subset *X* of the vertex set $V(G)$, the subgraph induced by *X* will be denoted by $G[X]$. A set $A\subseteq V$ is called a *module* if every vertex in $V-A$ is either adjacent to all vertices in *A* or none of them. The *representative* graph of a graph *G* is the subgraph of *G* induced by the set of vertices containing one vertex from each proper maximal module of *G*. A graph *G* is called *Z*-free where *Z* is a set of graphs, when *G* does not contain an induced subgraph isomorphic to a graph of *Z*.

Definition 1 [2]. *Given a bipartite graph* $G=(B\cup W,E)$ *of order at least* 2, *G is* $K+S$ *graph if and only if G contains an isolated vertex or* its *vertex set can be decomposed into two sets K and S such that K induces a complete bipartite graph while S is a stable set.*

Property 2 [2]. *Let* $G=(B\cup W,E)$ *be a bipartite graph of order at least* 2. *G is* $K+S$ *graph if and only if there exists a partition of its vertex set into two non empty classes* V_1 *and* V_2 *such that all possible edges exists between the black vertices of* V_1 *and the white vertices of* V_2 *while there is no edge connecting a white vertex of* V_1 *with a black vertex of* V_2.

Such partition is referred as associated partition of *G* and is denoted by the ordered pair (V_1, V_2) [2].

Property 3 [2]. *A bipartite graph G is a* $K+S$ *graph if and only if G admit a unique* (*up to isomorphism*) *partition of its vertex set* $(V_1\cup V_2\cup\cdots\cup V_k)$ *satisfying the following conditions*:

1) $\forall i=1,\cdots,k-1,(V_1\cup\cdots\cup V_i,V_{i+1}\cup\cdots\cup V_k)$ *is an associated partition to the graph G.*
2) $\forall i=1,\cdots,k, G[V_i]$ *is not a* $K+S$ *graph.*

The partition $(V_1,\cdots,V_k)$ of the above property is called $K+S$ decomposition while a set V_i said to be $K+S$ component of the graph.

From $K+S$ decomposition together with the decomposition of bipartite graph *G* into its connected components (parallel decomposition) or those of $\bar{G}^{bip}$ (series decomposition) yield a new decomposition scheme for *G* called *canonical decomposition*. It is shown in [2] that whatever the order in which the decomposition operators are applied ($K+S$ decomposition, series decomposition or parallel decomposition), a unique set of indecomposable (or prime) graphs with respect to canonical decomposition is obtained. Obviously, a unique tree is associated to this decomposition. The internal nodes are labeled according to the type of decomposition applied, while every leaf correspond to a vertex of *G*. Hence there are four types of internal nodes, parallel node (labeled *P*), series node (labeled *S*), $K+S$ node (labeled $K+S$), and indecomposable node (labeled *N*). By convention, the set of vertices corresponding to the set of leafs having an internal node α as their least common ancestor will be denoted simply by α.

Lozin in [1] gives the following characterization for bipartite $Star_{123}$-free graphs.

Theorem 4 [1]. *Let G be a bipartite* $Star_{123}$*-free graph. One of the following hold.*
1) *G is* $K+S$ *graph.*
2) *G and* $\overline{G}^{bip}$ *aren't both connected.*
3) *The representative graph of G or the bi-complement of the representative graph of G is a path* P_k *or a cycle* C_k *with* $k \geq 7$.

It is shown in [14] that the representative graph of a graph *G* is a path P_k or $\overline{P}_k^{bip}$ or a cycle C_k or $\overline{C}_k^{bip}$ if and only if *G* is an extended path EP_k or a bi-complement of an extended path EP_k or an extended cycle EC_k or a bi-complement of an extended cycle EC_k respectively. More precisely, (see **Figure 2**).

Definition 5 [14]. *A* graph *G is said to be an extended path* EP_k *if there is a partition of the vertex set of G into a monochromatic sets* $\{V_1,\cdots,V_k\}$ such *that* $E=\bigcup_{i=1}^{k-1} V_i \times V_{i+1}$.

Definition 6 [14]. *A graph G is said to be an extended cycle* EC_k *if there is a partition of the vertex set of G into a monochromatic sets* $\{V_1,\cdots,V_k\}$ *such that* $E=\bigcup_{i=1}^{k-1} V_i \times V_{i+1} \cup V_1 \times V_k$.

The construction of the canonical decomposition tree of a bipartite $Star_{123}$-free graph can be obtained in linear time from the algorithm given by Quaddoura in [14]. According to this algorithm, every child of a *N*-node is a node marked by P' corresponding to a set $V_i, i=1,\cdots,k$, if $|V_i|>1$, or to a vertex of *G* otherwise. **Figure 3** illustrates a bipartite $Star_{123}$-free graph and its canonical decomposition tree.

3. Maximum Matching of Bipartite *Star*₁₂₃-Free Graphs

In this section we will extend the techniques developed in [2] to provide an $O(n)$ time algorithm for the maximum matching problem on bipartite $Star_{123}$-free graph. We present first the required tools for this purpose.

A classical tool for solving the maximum matching problem was introduced by Berge in [16]. Let *M* be any matching of a graph $G=(V,E)$, an *M*-alternating path is a path whose edges are alternately in *M* and in $E-M$. If some edge of *M* is incident to a vertex *v*, this vertex is said to be *saturated* by *M*, otherwise *v* is *M-unsaturated*. An *M-augmenting* path is an *M*-alternating path whose both endpoints are *M*-unsaturated.

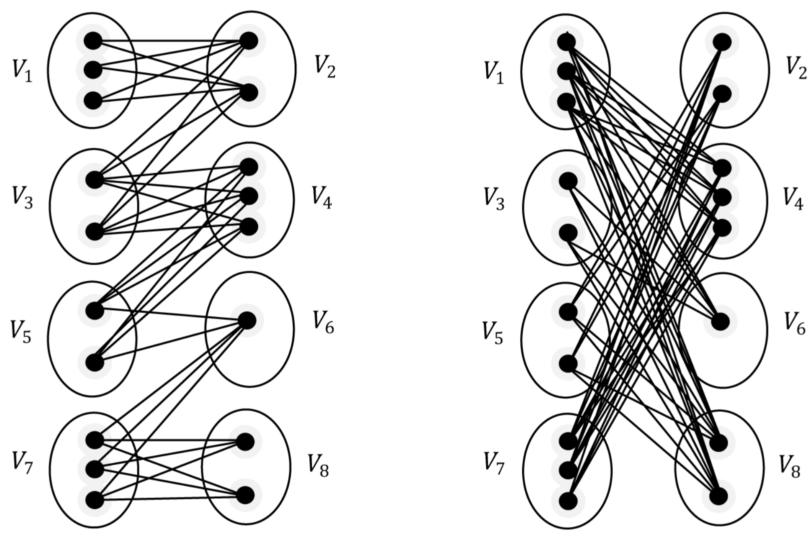

Figure 2. An EP_8 and an $\overline{EP}_8^{bip}$.

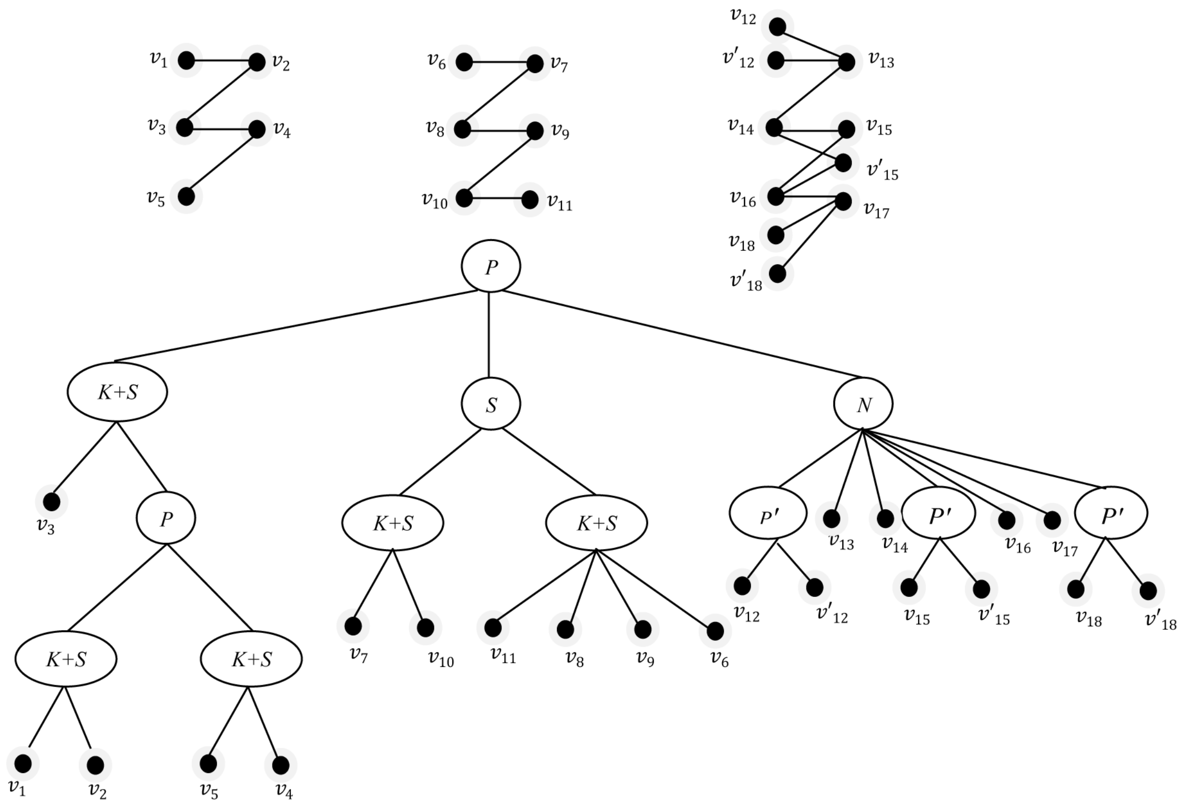

Figure 3. A bipartite $Star_{123}$-free graph and its canonical decomposition tree.

Theorem 7 [16]. *A matching M of a graph G is maximum matching if and only if G contain no M-augmenting path.*

Consider a bipartite graph G such that G admit a decomposition according to some rule into two graphs G_1 and G_2. Let M_1 and M_2 be maximum matchings of G_1 and G_2, let $M = M_1 \cup M_2$ which is a matching of G. In order to increase the size of M we use the operations Match and Split (see [11]) described below.

Let U_1 be the set of M_1-unsaturated vertices of G_1 and U_2 be the set of M_2-unsatureted vertices of G_2. A *Match operation* occurs if there are two adjacent vertices $v_1 \in U_1$ and $v_2 \in U_2$ then the edge $v_1 v_2$ is added to M, the vertices v_1 and v_2 are thus saturated by M and they are respectively deleted from the sets U_1 and U_2.

Let U be the set of M-unsaturated vertices, a *Split operation* occurs if there exists an edge of M say xy and vertices u and v belonging to U such that u is adjacent to x and v is adjacent to y. In that case the Split operation constructs a new matching M' defined by $M' = M \cup \{ux, vy\} - \{xy\}$, the vertices u and v being saturated by M' and deleted from U and the edge xy is deleted from M. Note that, if $G = (B \cup W, E)$ is a bipartite complete then a maximum matching of G can be obtained by applying Match operations between the two sets B and W.

Let now G be a bipartite $Star_{123}$-free graph and $T(G)$ is its canonical decomposition tree. For our purpose we shall modify $T(G)$ to a binary tree $BT(G)$ as follows: We visit all nodes of $T(G)$ in DFS order. For a node α of type P, S or $K+S$ let $\alpha_1, \cdots, \alpha_k$ be the children of α. If $k = 2$ then α does not change. Else α_1 remains its left child and $\acute{\alpha}$ is its new child labeled by P, S or $K+S$ respectively with $\alpha_2, \cdots, \alpha_k$ are its children. For a N-node α, using the Procedure MAXMATCH EP_k, EC_k, the Procedure MAXMATCH $\overline{EP}_k^{bip}$, or the procedure MAXMATCH $\overline{EC}_k^{bip}$ described below, which find a maximum matching of an extended path EP_k or an extended cycle EC_k or their bi-complements, we replace α by a leaf $\acute{\alpha}$ together with a maximum matching of the subgraph $G[\alpha]$ and the set of unsatureted vertices with respect to this matching. Our algorithm uses post order traversal to visit all the nodes of $BT(G)$. Whenever an internal node α of this binary tree is visited, we compute a maximum matching of $G[\alpha]$ from the maximum matching M_1

of $G[\alpha_1]$ and M_2 of $G[\alpha_2]$ where α_1 and α_2 are the two children of α in $BT(G)$. For this purpose we distinguish the following cases according to the type of α.

3.1. *α* Is of Type *P*, *S* or *K* + *S*

Consider now the set $M = M_1 \cup M_2$ which is a matching of $G[\alpha]$. Obviously, if α is a *P*-node then *M* is a maximum matching of $G[\alpha]$. In the case when α is of type *S* or *K* + *S* we use the same technique used in [6] to find a maximum matching of $G[\alpha]$.

Let U_1 be the set of M_1-unsaturated vertices of $G[\alpha_1]$ and U_2 be the set of M_2-unsatureted vertices of $G[\alpha_2]$. Let $Match[U_1, U_2]$ be the matching of $G[\alpha]$ obtained when all possible Match operations have been sequentially performed.

Let now *U* be the set of *M*-unsaturated vertices, and Let $Split[U, M]$ be the matching obtained when all possible Split operations have been sequentially performed.

Theorem 8 [2]. *If* α *is a K + S-node, the set* $Match[U_1, U_2]$ *is a* maximum *matching of* $G[\alpha]$.

Theorem 9 [2]. *Assume that* α *is a S-node and M is equal to* $Match[U_1, U_2]$. *Let U be the set of M-unsaturated vertices of* $G[\alpha]$, *then the set* $Split[U, M]$ *is a maximum matching of* $G[\alpha]$.

3.2. *α* Is of Type *N*

In this section we will develop an $O(n)$ algorithm to find a maximum matching of an extended path EP_k or an extended cycle EC_k and an $O(n)$ algorithm to find a maximum matching of their bi-complement (see Definitions 5 and 6). We can suppose that $k = 2n'$ if *k* is even or $k = 2n' + 1$ if *k* is odd. We denote by $Match[V_i, V_j]$ the matching of the bipartite complete graph $G[V_i \cup V_j]$ obtained by Match operations between the two monochromatic sets V_i and V_j. When an edge *xy* is added to this matching where $x \in V_i$ and $y \in V_j$ then *x* will be deleted from V_i and *y* will be deleted from V_j. Note that, during the execution of Procedure MAXMATCH EP_k, EC_k or the Procedures MAXMATCH $\overline{EP}_k^{bip}$ and MAXMATCH $\overline{EC}_k^{bip}$, the matching $Match[V_i, V_j]$ is not necessarily maximum for $G[V_i \cup V_j]$, this is because some vertices of V_i or V_j may already be saturated.

3.2.1. *α* Is EP_k or EC_k

Procedure MAXMATCH EP_k, EC_k provides a maximum matching of an extended path EP_k or an extended cycle EC_k. By convention, every monochromatic set of an extended path or an extended cycle has an odd index consists of black vertices and those having an even index consist of white ones. For the purpose of simplification, the length *k* of the extended path in this Procedure is considered to be odd, if this length is even then the set $V_{2n'+1}$ is considered to be empty.

Procedure MAXMATCH EP_k, EC_k

1) $M \leftarrow \varnothing$
2) if $G = EC_k$ then $M \leftarrow M \cup Match[V_1, V_k]$
3) for $i = 1$ to n' do
   begin for
4) $M \leftarrow M \cup Match[V_{2i-1}, V_{2i}]$
5) $M \leftarrow M \cup Match[V_{2i}, V_{2i+1}]$
   end for

Theorem 10. *Let* $G = (V, E)$ *be an extended path* EP_k *or an extended cycle* EC_k *where* $V = V_1 \cup V_2 \cup \cdots \cup V_{2n'+1}$. *Procedure* MAXMATCH EP_k, EC_k *produces a maximum matching of G.*

Proof. Let $P = v_1 v_2 \cdots v_t$ be an *M*-augmenting path in *G*. Since *G* is a bipartite, *t* is even, so v_1 and v_t are of different colors. Without loss of generality, assume that v_1 is a black vertex and v_t is white. Let $v_1 \in V_{2j-1}$ $(1 \le j \le n' + 1)$ and $v_t \in V_{2r}$ $(1 \le r \le n')$.

Claim 1. There is no black vertex of *P* in V_{2j+1}.

Proof. Let v_s be the first black vertex of *P* in V_{2j+1}, then v_{s-1} must be in V_{2j}. Since v_1 is a black vertex non saturated, it must be $v_{s-1} v_s \in M$. According to our Procedure, the edge $v_{s-1} v_s$ has been added to *M* by the operation $Match[V_{2j}, V_{2j+1}]$, but before this step, the edge $v_1 v_{s-1}$ must be added to *M* by the step

$Match\left[V_{2j-1},V_{2j}\right]$, a contradiction. ■

Claim 2. There is no white vertex of *P* in V_{2r+2}.

Proof. Let v_s be the last white vertex of *P* in V_{2r+2}, then v_{s+1} must be in V_{2r+1} or in V_{2r+3} or in V_1 and $v_s v_{s+1} \in M$. The vertex v_{s+1} does not belong to V_{2r+3}, otherwise, since $v_t \in V_{2r}$, the path $v_{s+1} \cdots v_t$ must contain a vertex in V_{2r+2}, a contradiction with our choice of v_s. If $v_{s+1} \in V_{2r+1}$ then according to our Procedure, the edge $v_s v_{s+1}$ has been added to *M* by the operation $Match\left[V_{2r+1},V_{2r+2}\right]$ when $i = r+1$, but in the step $i = r$, the edge $v_t v_{s+1}$ must be added to *M* by the operation $Match\left[V_{2r},V_{2r+1}\right]$, a contradiction. If $v_{s+1} \in V_1$ then $G = EC_k$ and $2r+2 = 2n'$. In this case, the vertex v_1 does not belong to $V_{2n'-1}$ or to $V_{2n'-3}$, otherwise $v_1 v_t$ must be added to *M*, so $j < n'-2$. But now the set V_{2j+1} must contain a vertex of *P*, a contradiction with Claim 1. ■

Suppose that *G* is an EP_k or *G* is an EC_k such that there is no edge of *P* connecting V_1 and V_k. If $j < r$ then either V_{2j+1} contains a vertex of *P* or there is an edge of *P* connecting V_1 and V_k, a contradiction. If $j > r$ then either V_{2r+2} contains a vertex of *P* or there is an edge of *P* connecting V_1 and V_k, a contradiction. Therefore $j = r$. But now the edge $v_1 v_t$ must be added to *M* by the operation $Match\left[V_{2r-1},V_{2r}\right]$, a contradiction.

Suppose now *G* is an EC_k and there is an edge of *P* connecting V_1 and V_k. Let $v_i v_{i+1}$ be the first edge of *P* connecting V_1 and V_k. By Claim 1 and Claim 2, $v_t \in V_k$. Thus v_1 does not belong to $V_1 \cup V_{k-1}$. If $v_i \in V_k$ then according to our choice of $v_i v_{i+1}$, the set V_{2j+1} must contain a vertex of *P*, a contradiction with Claim 1. Therefore $v_i \in V_1$. Since the vertex v_1 is a black non-saturated vertex, the edge $v_i v_{i+1}$ does not belong to *M*, By our choice of the edge $v_i v_{i+1}$, the vertex v_{i-1} must belong to V_2. Now, the edge $v_{i-1} v_i$ has been added to *M* by the operation $Match\left[V_1,V_2\right]$. But before this step, the edge $v_i v_t$ must be added to *M* by the operation $Match\left[V_1,V_k\right]$, a contradiction. □

The following **Table 1** illustrates a trace of the Procedure MAXMATCH EP_k, EC_k for the EP_8 in **Figure 2** where a vertex in V_i is denoted by $v_i^j, i = 1,\cdots,8,\ j = 1,\cdots,\left|V_i\right|$.

3.2.2. α Is $\overline{EP}_k^{bip}$ or $\overline{EC}_k^{bip}$

Note that the matching obtained by the Procedure MAXMATCH EP_k, EC_k is ensured to be maximum because of the order of applying Match operations. In the case when α is $\overline{EP}_k^{bip}$ or $\overline{EC}_k^{bip}$, we must also design an order of applying Match operations and Split operations to ensure that the resulting matching is maximum. For this purpose, we will study first the structure of a *M*-augmenting path of a matching *M* of α obtained by doing in an arbitrary order all possible Match operations then all possible Split operations (Lemma 12 and Lemma 13). Knowing this structure will enable us to design an order of applying Match operations (Procedure Match (*G*)) then developing a Procedure of a maximum matching *M* for $\overline{EP}_k^{bip}$ or $\overline{EC}_k^{bip}$.

Recall that when an edge *xy* is added to a matching *M* by a Match operation $Match\left[V_i,V_j\right]$ where $x \in V_i$ and $y \in V_j$ then *x* will be deleted from V_i and *y* will be deleted from V_j. In addition, we suppose here that Match operation associates labels with *x* and *y* as $Label(x) = i$ and $Label(y) = j$ when $x \in V_i$ and $y \in V_j$ respectively. Two monochromatic sets V_i and V_j of different color are called *independent* if $V_i \cup V_j$ form a stable set, *non-independent* otherwise.

Lemma 11. *Let* $G = \overline{EP}_k^{bip}$ *or* $G = \overline{EC}_k^{bip}$, *let M be a matching of G obtained when all possible Match operations have been performed. If there are at least two M-unsaturated vertices of different color then all the M-unsaturated vertices are* located *in at most three consecutive monochromatic sets* V_s, V_{s+1} *and* V_{s+2} *where* $s \in \{1,\cdots,k\}$.

Table 1. Illustration of procedure MAXMATCH $EP_k, EC_k S$ for the EP_8 in **Figure 2**.

i	$Match\left[V_{2i-1},V_{2i}\right]$	$Match\left[V_{2i},V_{2i+1}\right]$	M
1	$\{v_1^1v_2^1, v_1^2v_2^2\}$	____	$\{v_1^1v_1^2, v_2^1v_2^2\}$
2	$\{v_3^1v_4^1, v_3^2v_4^2\}$	$\{v_4^3v_5^1\}$	$\{v_1^1v_1^2, v_2^1v_2^2, v_3^1v_4^1, v_3^2v_4^2, v_4^3v_5^1\}$
3	$\{v_5^2v_6^1\}$	____	$\{v_1^1v_1^2, v_2^1v_2^2, v_3^1v_4^1, v_3^2v_4^2, v_4^3v_5^1, v_5^2v_6^1\}$
4	$\{v_7^1v_8^1, v_7^2v_8^2\}$	____	$\{v_1^1v_1^2, v_2^1v_2^2, v_3^1v_4^1, v_3^2v_4^2, v_4^3v_5^1, v_5^2v_6^1v_7^1v_8^1, v_7^2v_8^2\}$

Proof. By the hypothesis of the Lemma, all the M-unsaturated vertices must be in independent sets. Obviously any three consecutive sets are independent and the maximum number of independent sets is three. □

Assume that $G=\overline{EP}_k^{bip}$ or $G=\overline{EC}_k^{bip}$ and M is a matching of G obtained when all possible Match operations have been performed (in an arbitrary order). The following Procedure determines the sets V_s, V_{s+1}, V_{s+2} which are the possible location of M-unsaturated vertices. Note that when $G=\overline{EC}_k^{bip}$, the sets V_k and V_1 are consecutive.

Procedure *M*-unsaturated vertices (*G*, *M*)

1) Find the small index $s\in\{1,\cdots,k\}$ for which $V_s\neq\varnothing$
2) if there is no such s then return M is maximum
   else
3) if $G=\overline{EP}_k^{bip}$ or $s\neq 1$ then
   //when $G=\overline{EP}_k^{bip}$ and $s=k-1$, V_{s+2} does not exist, when $G=\overline{EC}_k^{bip}$ and $s=k-1$, $V_{s+2}=\varnothing$
4) if $s=k-1$ then return $V_s=V_{k-1}$ and $V_{s+1}=V_k$
5) else return V_s, V_{s+1}, V_{s+2}
   else $//G=\overline{EC}_k^{bip}$ and $s=1$
6) if $V_k=\varnothing$ then return $V_s=V_1, V_{s+1}=V_2, V_{s+2}=V_3$
7) else if $V_{k-1}=\varnothing$ then return $V_s=V_k, V_{s+1}=V_1, V_{s+2}=V_2$
8) else return $V_s=V_{k-1}, V_{s+1}=V_k, V_{s+2}=V_1$

According to Lemma 11, one of the two M-unsaturated vertices of any M-augmenting path in G is in V_s and the second in V_{s+1}, or one in V_{s+1} and the second in V_{s+2}. Consider first a M-augmenting path in G which its M-unsaturated vertices are in V_s and in V_{s+1}.

To augment the size of M, Split operations can be done between the M-unsaturated vertices of V_s, the M-unsaturated vertices of V_{s+1} and the edges of M whose extremities belong to monochromatic sets non-independent of V_s and V_{s+1}, namely the edges of M whose extremities don't belong to V_{s-1}, V_s, V_{s+1} when $G=\overline{EP}_k^{bip}$ and $s=k-1$, and the edges of M whose extremities don't belong to V_{s-1}, V_s, V_{s+1} and V_{s+2} otherwise. The following Procedure performs these Split operations.

Procedure Split (*M*, V_s, V_{s+1})

1) if $G=\overline{EP}_k^{bip}$ and $s=k-1$ then
   $M'=\{xy\in M \mid label(x) \text{ and } label(y)\neq s-1, s \text{ and } s+1\}$
2) else $M'=\{xy\in M \mid label(x) \text{ and } label(y)\neq s-1, s, s+1 \text{ and } s+2\}$
3) while $V_s\neq\varnothing$ and $V_{s+1}\neq\varnothing$ and $M'\neq\varnothing$ do
   Begin while
4) let $u\in V_s, v\in V_{s+1}, xy\in M', M=M-\{xy\}\cup\{ux, vy\}$
5) $V_s=V_s-\{u\}, V_{s+1}=V_{s+1}-\{v\}, M'=M'-\{xy\}$
   // assuming that u and x also v and y are of different color
   end while

The following Lemma describes the structure of a M-augmenting path whose extremities belong to V_s and V_{s+1}.

Lemma 12. *After the execution of* Procedure *Split* (M, V_s, V_{s+1}) *if there is a M-augmenting path* $P=v_1\cdots v_t$ *in G whose extremities in* V_s *and* V_{s+1} *then*:

- $G\neq\overline{EP}_k^{bip}$ *or* $s\neq k-1$.
- *P can be reduced to a M-augmenting path* $v_1v_2v_3v_4v_5v_6$ *where* $v_1\in V_s, v_2\in V_i, v_3\in V_{s+2}, v_4\in V_{s-1}, v_5\in V_j$, $v_6=v_t\in V_{s+1}$, V_i *is any non-independent set of* V_s *and* V_j *is any non-independent set of* V_{s+1}.

Proof. Let $P=v_1v_2\cdots v_t$ be a M-augmenting path in G where $v_1\in V_s$ and $v_t\in V_{s+1}$. Since after the execution of Procedure Split (M, V_s, V_{s+1}), $V_s\neq\varnothing$ and $V_{s+1}\neq\varnothing$, the set M' must be empty. Therefore, if $G=\overline{EP}_k^{bip}$ and $s=k-1$, every edge of M has an extremity in V_{s-1}, V_s or V_{s+1}, and if $G\neq\overline{EP}_k^{bip}$ or $s\neq k-1$, every edge of M has an extremity in V_{s-1}, V_s, V_{s+1} or V_{s+2}. Obviously, the color of every vertex of P having an odd index (resp. even index) is as the color of v_1 (resp. v_t).

Let v_i be the first vertex of P having an odd index and belongs to a set distinct of V_s, then $v_{i-1}v_i\in M$.

Since $M' = \varnothing$ and $v_i \notin V_s$, either $v_{i-1} \in V_{s-1} \bigcup V_{s+1}$ or $v_i \in V_{s+2}$. If $v_{i-1} \in V_{s-1} \bigcup V_{s+1}$ then $v_{i-2} \notin V_s$, a contradiction with our choice of v_i, therefore $v_i \in V_{s+2}$. Since V_{s+2} does not exist when $G = \overline{EP}_k^{bip}$ and $s = k-1$, then $G \neq \overline{EP}_k^{bip}$ or $s \neq k-1$. Since $v_{i-1} \notin V_{s-1} \bigcup V_{s+1}$, $v_1 v_{i-1} \in E(G)$. Now the subpath $v_1 v_2 \cdots v_i$ of P can be reduced to $v_1 v_{i-1} v_i$.

Let v_j be the last vertex of P having an even index and belongs to a set distinct of V_{s+1}, then $v_j v_{j+1} \in M$. Since $M' = \varnothing$ and $v_j \notin V_{s+1}$, either $v_{j+1} \in V_s \bigcup V_{s+2}$ or $v_j \in V_{s-1}$. If $v_{j+1} \in V_s \bigcup V_{s+2}$ then $v_{j+2} \notin V_{s+1}$, a contradiction with our choice of v_j, therefore $v_j \in V_{s-1}$. Since $v_i \in V_{s+2}$ and $v_j \in V_{s-1}$, $v_i v_j \in E(G)$. Obviously $v_i v_j \notin M$. Since $v_{j+1} \notin V_s \bigcup V_{s+2}$, $v_{j+1} v_t \in E(G)$. Now the path P can be reduced to the M-augmenting path $v_1 v_{i-1} v_i v_j v_{j+1} v_t$. □

Consider now a M-augmenting path in G such that its M-unsaturated vertices are in V_{s+1} and in V_{s+2}. In a similar way, by replacing in the above Procedure and in Lemma 12, $s-1$ by s, s by $s+1$, $s+1$ by $s+2$ and $s+2$ by $s+3$, we obtain the Procedure Split (M, V_{s+1}, V_{s+2}) and Lemma 13which describes the structure of a M-augmenting path whose extremities belong to V_{s+1} and V_{s+2}.

Procedure Split (M, V_{s+1}, V_{s+2})

1) if $G = \overline{EP}_k^{bip}$ and $s+1 = k-1$ then
$M'' = \{xy \in M \mid label(x) \text{ and } label(y) \neq s, s+1, s+2\}$
2) else $M'' = \{xy \in M \mid label(x) \text{ and } label(y) \neq s, s+1, s+2 \text{ and } s+3\}$
3) while $V_{s+1} \neq \varnothing$ and $V_{s+2} \neq \varnothing$ and $M'' \neq \varnothing$ do
begin while
4) let $u \in V_{s+1}, v \in V_{s+2}, xy \in M'', M = M - \{xy\} \bigcup \{ux, vy\}$
5) $V_{s+1} = V_{s+1} - \{u\}, V_{s+2} = V_{s+2} - \{v\}, M'' = M'' - \{xy\}$
//assuming that u and x also v and y are of different color
end while

Lemma 13. *After the execution of Procedure Split* (M, V_{s+1}, V_{s+2}) *if there is a M-augmenting path* $P = v_1 \cdots v_t$ *in G whose extremities in* V_{s+1} *and* V_{s+2} *then*:

- $G \neq \overline{EP}_k^{bip}$ *or* $s+1 \neq k-1$.
- *P can be reduced to a M-augmenting path* $v_1 v_2 v_3 v_4 v_5 v_6$ *where* $v_1 \in V_{s+1}, v_2 \in V_i, v_3 \in V_{s+3}, v_4 \in V_s, v_5 \in V_j$, $v_6 = v_t \in V_{s+2}$, V_i *is any non-independent set of* V_{s+1} *and* V_j *is any non-independent set of* V_{s+2}.

We start now by developing a Procedure for a maximum matching in $G = \overline{EP}_k^{bip}$ or $G = \overline{EC}_k^{bip}$. The order of applying Match operations is defined in following Procedure which called MATCH (G). Recall that either $k = 2n'$ or $k = 2n'+1$.

Procedure MATCH (G)

1) $M \leftarrow \varnothing$
2) $l = 2, h = 1$
3) For $i = 2$ to n' (or to $n'+1$ if $G = \overline{EP}_{2n'+1}^{bip}$) do
begin for
4) $j = l$
5) while $V_{2i-1} \neq \varnothing$ and $j \leq 2i-4$ do
begin while
6) $M \leftarrow M \bigcup Match[V_{2i-1}, V_j]$
7) if $V_j = \varnothing$ then $j = j+2$
end while
8) $l = j$
9) $j = h$
10) while $V_{2i} \neq \varnothing$ and $j \leq 2i-3$ do
begin while
11) $M \leftarrow M \bigcup Match[V_{2i}, V_j]$
12) if $V_j = \varnothing$ then $j = j+2$
end while
13) $h = j$
end for

Procedure MATCH (*G*) works as following, for every $i = 2$ to n' (or to $n'+1$ if $G = \overline{EP}^{bip}_{2n'+1}$):

- Add to *M* the possible edges between V_{2i-1} as long as it is non empty and (the non-independent sets of V_{2i-1} having indices less than $2i-1$) $V_l, V_{l+2}, \cdots, V_{2i-4}$ with respect to this order, where *l* determines the last non empty set in $V_2, \cdots, V_{2i-2}$ during the for loop iterations $2, \cdots, i-1$.
- Add to *M* the possible edges between V_{2i} as long as it is non empty and (the non-independent sets of V_{2i} having indices less than 2*i*) $V_h, V_{h+2}, \cdots, V_{2i-3}$ with respect to this order, where *h* determines the last non empty set in $V_1, \cdots, V_{2i-1}$ during the for loop iterations $2, \cdots, i-1$.

Observation 14. According to Procedure MATCH (*G*):

- if xy is an edge of *M* created by the operation $Match[V_i, V_j]$ then $i > j$.
- if $i < i'$ (resp. $j < j'$) then the edges of $Match[V_i, V_j]$ have been added to *M* before adding the edges of $Match[V_{i'}, V_j]$ (resp. $Match[V_i, V_{j'}]$).
- if $i > j > r$ then the edges of $Match[V_j, V_r]$ have been added to *M* before adding the edges of $Match[V_i, V_j]$.

The following **Table 2** illustrates a trace of the Procedure MATCH (*G*) for the $\overline{EP}^{bip}_8$ in **Figure 2**. The second and the third column of this table represent the execution of steps 5 and 10 respectively.

The combination of Procedures MATCH (*G*), *M*-unsaturated vertices (G, M), Split (M, V_s, V_{s+1}), and Split (M, V_{s+1}, V_{s+2}) provides the Procedure MAXMMATCH $\overline{EP}^{bip}_k$. For a maximum matching of $\overline{EC}^{bip}_k$ we need a little addition. Theorem 15 proves their correctness.

Procedure MAXMATCH $\overline{EP}^{bip}_k$

1) MATCH $(\overline{EP}^{bip}_k)$
2) *M*-unsaturated vertices $(\overline{EP}^{bip}_k, M)$
3) Split (M, V_s, V_{s+1})
4) Split (M, V_{s+1}, V_{s+2})

Procedure MAXMATCH $\overline{EC}^{bip}_k$

1) MATCH $(\overline{EC}^{bip}_k)$
2) *M*-unsaturated vertices $(\overline{EC}^{bip}_k, M)$
3) Split (M, V_s, V_{s+1})
4) Split (M, V_{s+1}, V_{s+2})
5) if $V_{k-1} \neq \varnothing$ and $V_k \neq \varnothing$ then
   //Assuming that *x* and the vertices of V_k also *y* and the vertices of V_{k-1} are of the same color
6) $M' = \{xy \in M \mid label(x) \neq k-2, label(y) = 1\} \neq \varnothing$
7) $M'' = \{xy \in M \mid label(x) = k-2, label(y) \neq 1\}$
8) while $V_{k-1} \neq \varnothing$ and $V_k \neq \varnothing$ and $M' \neq \varnothing$ and $M'' \neq \varnothing$ do
   begin while
   let $u \in V_{k-1}, v \in V_k, xy \in M', x'y' \in M''$
9) $M \leftarrow M - \{xy, x'y'\} \cup \{ux, x'y, vy'\}$
10) $V_{k-1} = V_{k-1} - \{u\}, V_k = V_k - \{v\}, M' = M' - \{xy\}, M'' = M'' - \{x'y'\}$
   end while

Theorem 15. *Procedure MAXMATCH $\overline{EP}^{bip}_k$ and Procedure MAXMATCH $\overline{EC}^{bip}_k$ produce a maximum matching of $G = \overline{EP}^{bip}_k$ and $G = \overline{EC}^{bip}_k$ respectively.*

Proof. Suppose that after execution of Procedure MAXMATCH $\overline{EP}^{bip}_k$ or Procedure MAXMATCH $\overline{EC}^{bip}_k$ there is a *M*-augmenting path $P = v_1 v_2 \cdots v_t$. Since v_1 and v_t are of different color and all the monochromatic sets are empty except at most V_s, V_{s+1} and V_{s+2}, there are two cases, either $v_1 \in V_s, v_t \in V_{s+1}$ or $v_1 \in V_{s+1}, v_t \in V_{s+2}$.

Let $r = s$ when $v_1 \in V_s, v_t \in V_{s+1}$ or $r = s+1$ when $v_1 \in V_{s+1}, v_t \in V_{s+2}$. By Lemma 12 and Lemma 13, *P* can be reduced to a path $v_1 v_2 v_3 v_4 v_5 v_6$ where $v_1 \in V_r, v_2 \in V_i, v_3 \in V_{r+2}, v_4 \in V_{r-1}, v_5 \in V_j, v_6 = v_t \in V_{r+1}$, V_i is any non-independent set of V_r and V_j is any non-independent set of V_{r+1}. Assume first that $r \neq k-1$ and $r \neq k$.

Claim 1. The edge $v_2 v_3$ is obtained by Split operation.

Proof. Suppose that the edge $v_2 v_3$ is obtained by Match operation. Since $v_3 \in V_{r+2}$ the edge $v_2 v_3$ is obtained either by $Match[V_{r+2}, V_i]$ or by $Match[V_i, V_{r+2}]$. Without loss of generality, assume that $v_2 v_3$ is ob-

Table 2. Illustration of Procedure MATCH (G) for the $\overline{EP}_8^{bip}$ in **Figure 2**.

i	h	l	j	$Match[V_{2i-1},V_j]$	j	$Match[V_{2i},V_j]$	M
2	2	2	2	____	1	$\{v_4^1v_1^1,v_4^2v_1^2,v_4^3v_1^3\}$	$\{v_4^1v_1^1,v_4^2v_1^2,v_4^3v_1^3\}$
			2	____	3	____	
3	2	3	2	$\{v_5^1v_2^1,v_5^2v_2^2\}$	3	$\{v_6^1v_3^1\}$	$\{v_4^1v_1^1,v_4^2v_1^2,v_4^3v_1^3,v_5^1v_2^1,v_5^2v_2^2,v_6^1v_3^1\}$
			4	____	3	____	
4	4	3	4	____	3	$\{v_8^1v_3^2\}$	$\{v_4^1v_1^1,v_4^2v_1^2,v_4^3v_1^3,v_5^1v_2^1,v_5^2v_2^2,v_1^6v_1^3,v_8^1v_3^2\}$
			6	____	6	____	

tained by $Match[V_{r+2},V_i]$. By Observation 14, $r+2>i$. Since $v_1 \in V_r$, $r>i$. So the operation $Match[V_r,V_i]$ exists and must precedes the operation $Match[V_{r+2},V_i]$ by Observation 14 and since $r \neq k-1,k$. Therefore, the edge v_1v_2 must be added to *M* instead of adding the edge v_2v_3, a contradiction. ■

Claim 2. $r=s$.

Proof. If $r=s+1$ then $v_3 \in V_{s+3}$. By Claim 1, v_2v_3 is obtained by Split operation, thus the vertex v_2 must belong to V_{s+2}, a contradiction since V_{s+2} and V_{s+3} are independent. ■

Since $v_3 \in V_{s+2}$, the edge v_2v_3 is obtained by the step 4, that is by Split (M,V_{s+1},V_{s+2}). Let $xy \in M''$ be the edge which was in *M* and which has been used in step 4 of the Procedure Split (M,V_{s+1},V_{s+2}) to obtain the edge v_2v_3. The vertex v_2 must be identical to *x* or to *y*. Let *x* be the vertex v_2. By the definition of M'', the vertices *x* and *y* don't belong to $V_s \cup V_{s+1} \cup V_{s+2}$. Since $v_1 \in V_s$ and $v_1v_2 \in E$, $x=v_2 \notin V_{s-1}$. Obviously, $y \notin V_{s-1}$. Therefore $xy \in M'$ where M' is the set defined in step 4 of the Procedure Split (M,V_s,V_{s+1}). But before executing the step Split (M,V_{s+1},V_{s+2}), the set M' must be empty since $V_s \neq \emptyset, V_{s+1} \neq \emptyset$, a contradiction.

Assume that $r=k-1$. Then $v_1 \in V_{k-1}$ and $v_t \in V_k$. By Lemma 12 and Lemma 13, $G=\overline{EC}_k^{bip}$, $v_1 \in V_{k-1}$, $v_2 \in V_i, v_3 \in V_1, v_4 \in V_{k-2}, v_5 \in V_j, v_6 = v_t \in V_k$, V_i is any non-independent set of V_{k-1} and V_j is any non-independent set of V_k. Since $v_1 \in V_{k-1}, v_6 \in V_k$ then $V_{k-1} \neq \emptyset$ and $V_k \neq \emptyset$. Therefore one of the sets M' and M'' in step 8 must be empty. This is contradicted with the fact that $v_2v_3 \in M'$ and $v_4v_5 \in M''$.

Assume finally that $r=k$. Then $v_1 \in V_k$ and $v_t \in V_1$. Since in this case V_k and V_1 must be independent, $G=\overline{EC}_k^{bip}$. By Lemma 12 and Lemma 13, $v_1 \in V_k, v_2 \in V_i, v_3 \in V_2, v_4 \in V_{k-1}, v_5 \in V_j, v_6 = v_t \in V_1$, V_i is any non-independent set of V_k and V_j is any non-independent set of V_1.

Claim 3. The edge v_4v_5 is obtained by Split operation.

Proof. Suppose that the edge v_4v_5 is obtained by Match operation. Since $k-1>j$, $v_4 \in V_{k-1}$ and $v_5 \in V_j$, then the edge v_4v_5 is obtained by $Match[V_{k-1},V_j]$. By Observation 14, since $k-1>j>1$, the operation $Match[V_j,V_1]$ exists and must precede the operation $Match[V_{k-1},V_j]$. Therefore, the edge v_5v_6 must be added to *M* instead of adding the edge v_4v_5, a contradiction. ■

Claim 4. $r=s+1$.

Proof. If $r=s$ then $v_4 \in V_{s-1}$. By Claim 3, v_4v_5 is obtained by Split operation, thus the vertex v_5 must belong to V_s, a contradiction since V_{s-1} and V_s are independent. ■

Since $v_4 \in V_s$, the edge v_4v_5 is obtained by the step 3, that is by Split (M,V_s,V_{s+1}). Let $xy \in M'$ be the edge which was in *M* and which has been used in step 4 of the Procedure Split (M,V_s,V_{s+1}) to obtain the edge v_4v_5. The vertex v_5 must be identical to *x* or to *y*. Let *x* be the vertex v_5 and let $y \in V_h$. By the definition of M', $h \neq 1$. Obviously, xy was not created by split operation. By Observation 14, if $j>h$ (resp. $j<h$) then xy was created by $Match[V_j,V_h]$ (resp. $Match[V_h,V_j]$). Since $1<h$ (resp. $h>j>1$), the operation $Match[V_j,V_1]$ precedes the operation $Match[V_j,V_h]$ (resp. $Match[V_h,V_j]$). So the edge v_5v_6 must be added to *M* instead of adding xy, a contradiction. □

Lets apply the Procedure MAXMATCH $\overline{EP}_k^{bip}$ on the graph $G=\overline{EP}_8^{bip}$ in **Figure 2**. As we shown above, Procedure MATCH (*G*) produces the matching $M=\{v_4^1v_1^1,v_4^2v_1^2,v_4^3v_1^3,v_5^1v_2^1,v_5^2v_2^2,v_1^6v_1^3,v_8^1v_3^2\}$. Procedure *M*-unsaturated vertices (G,M) gives that $V_s=V_{k-1}=V_7, V_{s+1}=V_k=V_8$ and V_{s+2} does not exist. Procedure Split (M,V_s,V_{s+1}) gives that $M'=\{v_4^1v_1^1,v_4^2v_4^2,v_4^3v_1^3,v_5^1v_2^1,v_5^2v_2^2\}$ and $M=\{v_4^1v_1^1,v_4^2v_1^2,v_4^3v_1^3,v_5^1v_2^1,v_8^2v_5^2,v_7^1v_2^1,v_6^1v_3^1,v_8^1v_3^2\}$. Since V_{s+2} does not exist, Procedure Split (M,V_{s+1},V_{s+2}) gives nothing.

3.3. The Whole Algorithm

Let us present now our algorithm for the maximum matching problem on bipartite $Star_{123}$-free graphs. Theorems 8, 9, 10, and 15 prove its correctness.

Algorithm Maximum Matching

Input: A bipartite $Star_{123}$-free graph G and its binary canonical decomposition tree $BT(G)$.
Output: M a maximum matching of G and U the set of M-unsaturated vertices of G.

1) Let α be a node on a postorder traversal of $BT(G)$.
2) If α is a leaf or α is a P'-node then $M \leftarrow \varnothing, U \leftarrow \{\alpha\}$.
3) Else if α is a N-node then.
4) If $G[\alpha] = EP_k$ or $G[\alpha] = EC_k$ then $M \leftarrow$ MAXMATCHEP_k, EC_k, $U \leftarrow M$-unsaturated vertices.
5) Else if $G = \overline{EP}_k^{bip}$ then $M \leftarrow$ MAXMATCH$\overline{EP}_k^{bip}$, $U \leftarrow M$-unsaturated vertices.
6) Else $M \leftarrow$ MAXMATCH$\overline{EC}_k^{bip}$, $U \leftarrow M$-unsaturated vertices.
7) Replace α by a leaf α' together with M and U.
8) Else let α_1 and α_2 be the two children of α in $BT(G)$.
9) Let M_1 and M_2 be respectively the maximum matchings and.
10) U_1 and U_2 be respectively the sets of unsaturated vertices of $G[\alpha_1]$ and $G[\alpha_2]$.
11) If α is a P-node then $M \leftarrow M_1 \cup M_2, U \leftarrow U_1 \cup U_2$.
12) Else if α is a $K+S$-node then $M \leftarrow Match[U_1, U_2]$, $U \leftarrow M$-unsaturated vertices.
13) Else $M \leftarrow Match[U_1, U_2]$, $U \leftarrow M$-unsaturated vertices $M \leftarrow Split[U_1, U_2]$, $U \leftarrow M$-unsaturated vertices.

3.4. Complexity

We show now that the complexity of our algorithm is $O(n)$.

The total number of Match operations performed by $Match[V_i, V_j]$, is at most $\min(|V_i|, |V_j|)$. So the run time of step 4 (Procedure MAXMATCH_EP_k_EC_k) is $O(n)$.

Consider the steps 5 and 6 which are the Procedures MAXMATCH $\overline{EP}_k^{bip}$ and MAXMATCH $\overline{EC}_k^{bip}$. The variables l and h in Procedure MATCH (G) assure that the sum of iterations of all while loops in this Procedure is $O(n')$. Since $n' < n$ and the number of Match operation performed by $Match[V_i, V_j]$ is at most $\min(|V_i|, |V_j|)$ then MATCH (G) runs in $O(n)$ time. The Procedure M-unsaturated vertices (G, M) runs in $O(n)$ time since $n' < n$.

Since the size of the matching obtained by MATCH (G) is less than or equal to $n/2$, the construction of the set M' and M'' defined in Procedures Split (M, V_s, V_{s+1}), Split (M, V_{s+1}, V_{s+2}) and MAXMATCH $\overline{EC}_k^{bip}$, as well as the while loops defined in these Procedures costs $O(n)$ time. So steps 5 and 6 runs in $O(n)$ time.

The total number of Match or Split operations performed in steps 8 to 13 is bounded by the size of maximum matching obtained, which is less or equal to $n/2$ ([2]), so the run time of steps 8 to 13 is $O(n)$.

Finally, since the number of visited nodes in $BT(G)$ is $O(n)$, this algorithms runs with $O(n)$ time complexity.

4. Conclusion

The maximum matching is computed in $O(n)$ time, given a binary canonical decomposition tree of a bipartite $Star_{123}$-free graph. The canonical decomposition of a bipartite $Star_{123}$-free graph can be done in $O(n+m)$ time [14] including the binary canonical decomposition tree construction. Thus, the whole process is in $O(n+m)$ time.

Acknowledgements

This research is funded by the Deanship of Research and Graduate Studies in Zarqa University/Jordan. The author is grateful to anonymous referee's suggestion and improvement of the presentation of this paper.

References

[1] Lozin, V.V. (2002) Bipartite Graphs without a Skew Star. *Discrete Mathematics*, **257**, 83-100. http://dx.doi.org/10.1016/S0012-365X(01)00471-X

[2] Fouquet, J.L., Giakoumakis, V. and Vanherpe, J.M. (1999) Bipartite Graphs Totally Decomposable by Canonical Decomposition. *International Journal of Foundation of Computer Science*, **10**, 513-533. http://dx.doi.org/10.1142/S0129054199000368

[3] Ben-Dor, A., Karp, R.M., Schwikowski, B. and Shamir, R. (2003) The Restriction Scaf-Fold Problem. *Journal of Computational Biology*, **10**, 385-398. http://dx.doi.org/10.1089/10665270360688084

[4] Buss, S.R. and Yianilos, P.N. (1995) A Bipartite Matching Approach to Approximate String Comparison an Search. Technical Report, NEC Research Institute, Princeton.

[5] Demirci, M.F., Shokoufandeh, A., Keselman, Y., Bretzner, L. and Dickinson, S. (2006) Object Recognition as Many-to-Many Feature Matching. *International Journal of Computer Vision*, **69**, 203-222. http://dx.doi.org/10.1007/s11263-006-6993-y

[6] Toussaint, G.T. (2004) A Comparison of Rhythmic Similarity Measures. 5*th International Conference on Music Information Retrieval*, 242-245.

[7] Toussaint, G.T. (2005) The Geometry of Musical Rhythm. *Japan Conference on Discrete and Computational Geometry*, Berlin-Heidelberg, 198-212. http://dx.doi.org/10.1007/11589440_20

[8] Micali and Vazirani, V.V. (1980) An $O\left(m\sqrt{n}\right)$ Algorithm for Finding Maximum Matching in General Graphs. *FOCS*, 17-27.

[9] Moitra and Johnson, R.C. (1989) A Parallel Algorithm for Maximum Matching in Interval Graphs. *Proceedings of International Conference on Parallel Processing*, **3**, 114-120.

[10] Alt, H., Blum, N., Mehlhborn, K. and Paul, M. (1991) Computing a Maximum Cardinality Matching in Bipartite Graphs in Time $O\left(n^{1.5}\sqrt{m\log n}\right)$. *Information Processing Letters*, **37**, 237-240. http://dx.doi.org/10.1016/0020-0190(91)90195-N

[11] Yu, M.S. and Yang, C.H. (1993) An O(n) Time Algorithm for Maximum Matching on Cographs. *Information Processing Letters*, **47**, 89-93. http://dx.doi.org/10.1016/0020-0190(93)90230-7

[12] Fouquet, J.L., Parfenoff, I. and Thuillier, H. (1997) An O(n) Time Algorithm for Maximum Matching in P4-Tidy Graphs. *Information Processing Letters*, **62**, 281-287. http://dx.doi.org/10.1016/S0020-0190(97)00081-1

[13] Quaddoura, R. (2014) An O(n) Time Algorithm for Maximum Induced Matching In Bipartite $Star_{123}$-Free Graphs. *World of Computer Science and Information Technology Journal* (*WCSIT*), **4**, 38-41.

[14] Quaddoura, R. (2006) Linear Time Recognition Algorithm of Bipartite $Star_{123}$-Free Graphs. *International Arab Journal of Information Technolog*, **3**, 193-202.

[15] Bondy, J.A. and Murty, U.S.R. (1979) Graph Theory with Applications. North Holland, New York.

[16] Berge, C. (1957) Two Theorems in Graph Theory. *Proceedings of the National Academy of Sciences of the United States of America*, **43**, 842-844. http://dx.doi.org/10.1073/pnas.43.9.842