Arc-disjoint in- and out-branchings in semicomplete split digraphs

Jiangdong Ai*, Yiming Hao†, Zhaoxiang Li‡, Qi Shao§

April 8, 2025

Dedicated to Professor Fuji Zhang on the occasion of his 88th birthday.

Abstract

An *out-tree (in-tree)* is an oriented tree where every vertex except one, called the *root*, has in-degree (out-degree) one. An *out-branching* B_u^+ *(in-branching* B_u^-*)* of a digraph D is a spanning out-tree (in-tree) rooted at u. A *good* (u, v)*-pair* in D is a pair of branchings B_u^+, B_v^- which are arc-disjoint. Thomassen proved that deciding whether a digraph has any good pair is NP-complete. A *semicomplete split digraph* is a digraph where the vertex set is the disjoint union of two non-empty sets, V_1 and V_2, such that V_1 is an independent set, the subdigraph induced by V_2 is semicomplete, and every vertex in V_1 is adjacent to every vertex in V_2. In this paper, we prove that every 2-arc-strong semicomplete split digraph D contains a good (u, v)-pair for any choice of vertices u, v of D, thereby confirming a conjecture by Bang-Jensen and Wang [Bang-Jensen and Wang, J. Graph Theory, 2024].

1 Introduction

The notation follows [2], so we only repeat a few definitions here. A digraph is not allowed to have parallel arcs or loops. A directed multigraph can have parallel arcs but no loops. A directed multigraph is called *semicomplete* if there is no pair of non-adjacent vertices. A directed multigraph D is *k-arc-strong* if $D \setminus A'$ remains strong for every subset $A' \subseteq A(D)$ of size at most $k - 1$.

An *out-branching* (resp., *in-branching*) of D is a spanning oriented tree in which every vertex except one, called the root, has in-degree (resp., out-degree) one in D. A *good* (u, v)*-pair* in D is a pair of arc-disjoint out-branching and in-branching rooted at u and v respectively. We say a vertex pair (u, v) is a *good vertex pair* if there is a good (u, v)-pair in D. Thomassen [11] proposed the following conjecture.

*Corresponding author. School of Mathematical Sciences and LPMC, Nankai University, Tianjin 300071, P.R. China. Email: jd@nankai.edu.cn. partially supported by the National Natural Science Foundation of China:12401456.
†School of Mathematical Sciences and LPMC, Nankai University, Tianjin 300071, P.R. China. Email: 1120230031@mail.nankai.edu.cn.

‡School of Mathematical Sciences and LPMC, Nankai University, Tianjin 300071, P.R. China. Email: ZhaoxiangLi@mail.nankai.edu.cn.

§Center for Combinatorics and LPMC, Nankai University, Tianjin 300071, P.R China. Email: Shaoqi@mail.nankai.edu.cn.

Conjecture 1.1. *[11] There exists an integer K such that every K-arc-strong digraph $D = (V, A)$ has a good (u, v)-pair for every choice of $u, v \in V$.*

A *strong arc decomposition* of a (multi-)digraph $D = (V, A)$ is a partition of its arc set A into two disjoint arc sets A_1 and A_2 such that both of the spanning subdigraphs $D_1 = (V, A_1)$ and $D_2 = (V, A_2)$ are strong. Clearly, if a digraph D has a strong arc decomposition, then D contains a good (u, v)-pair for every choice of $u, v \in V$. Therefore, the following conjecture by Bang-Jensen and Yeo would imply Conjecture 1.1.

Conjecture 1.2. *[10] There exists an integer K such that every K-arc-strong digraph has a strong arc decomposition.*

We can easily see that every digraph D with a strong arc decomposition is 2-arc-strong. Then, asking if every 2-arc-strong digraph has a strong arc decomposition is natural. Unfortunately, the following digraphs provide a negative answer.

Bang-Jensen and Yeo [10] proved that for a 2-arc-strong semicomplete digraph D, there is only one exception that does not have a strong arc decomposition, which is the digraph S_4 depicted in Figure 1. Bang-Jensen, Gutin and Yeo [5] generalized the above result to semicomplete multi-digraphs with six additional exceptions, see Figure 1.

Theorem 1.3. [1] *[5] A 2-arc-strong semicomplete multi-digraph $D = (V, A)$ on n vertices has a strong arc decomposition if and only if D is not isomorphic to one of the exceptional digraphs depicted in Figure 1. Furthermore, a strong arc decomposition of D can be obtained in polynomial time when it exists.*

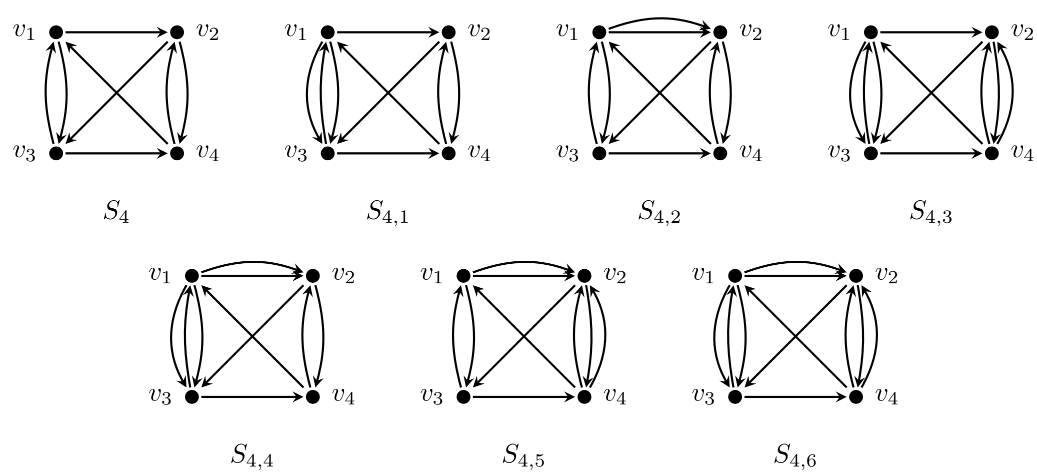


Figure 1: 2-arc-strong multi-digraphs without strong arc decompositions.

Subsequently, Bang-Jensen and Huang [6] extended semicomplete digraphs to locally semicomplete digraphs. Bang-Jensen, Gutin and Yeo [5] considered the strong arc decomposition of semicomplete composition and solved it completely.

[1] This is a modified version, the authors of [5] missed $S_{4,4}, S_{4,5}$ and $S_{4,6}$.

A *split digraph* D is a digraph whose vertex set is a disjoint union of two non-empty sets V_1 and V_2 such that V_1 is an independent set, the subdigraph induced by V_2 is semicomplete, and we denote it as $D = (V_1, V_2; A)$. Additionally, if every vertex in V_1 is adjacent to every vertex in V_2, we call such a split digraph *semicomplete split digraph*. Recently, Bang-Jensen and Wang [9] explored the strong arc decomposition of split digraphs. Their main result is the following:

Theorem 1.4. *[9] Let $D = (V_1, V_2; A)$ be a 2-arc-strong split digraph such that V_1 is an independent set and the subdigraph induced by V_2 is semicomplete. If every vertex of V_1 has both out- and in-degree at least 3 in D, then D has a strong arc decomposition.*

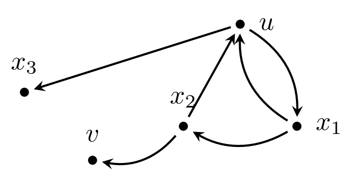
Later, Ai, He, Li, Qin and Wang [2] provided a complete characterization of whether a split digraph has a strong arc decomposition.

Theorem 1.5. *[2] A 2-arc-strong split digraph $D = (V_1, V_2; A)$ has a strong arc decomposition if and only if D is not isomorphic to any of the digraphs illustrated as follows, or their arc-reversed versions (reverse all arcs).*

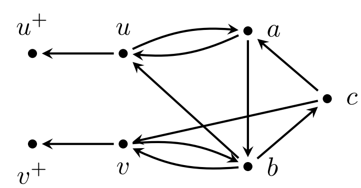
(1). *There are $x_1, x_2, u \in V(D)$ such that $N_D^+(u) = \{x_1, x_3\}$, $N_D^-(u) = \{x_1, x_2\}$, $N_D^+(x_1) = \{x_2, u\}$, $N_D^+(x_2) = \{v, u\}$, where $x_3, v \in V(D)\backslash\{x_1, x_2, u\}$ and x_3, v can be the same.*

(2). *There exist $a, b, c \in V_2, u, v \in V_1$, such that $N_D^+(b) = \{u, v, c\}, N_D^+(c) = \{v, a\}, N_D^+(a) = \{u, b\}, N_D^+(u) = \{a, u^+\}, N_D^+(v) = \{b, v^+\}$, where $u^+, v^+ \in V_2 \setminus \{a, b, c\}$ and u^+, v^+ can be the same, besides, $N_D^-(u) = \{a, b\}, N_D^-(v) = \{b, c\}$.*

(3). *The Appendix in [2].*



An illustration of counterexample in **(1)**



An illustration of counterexample in **(2)**

Note that if a digraph D contains a strong arc decomposition, it has a good (u, v)-pair for every choice of vertices $u, v \in D$. From Theorem 1.5, we see the condition 2-arc-strong is not enough to guarantee a strong arc decomposition for a split digraph, then what if a good (u, v)-pair for every choice of u, v? In [9], Bang-Jensen and Wang proved that there are infinitely many 2-arc-strong split digraphs that do not have good (u, v)-pairs for some choice of u, v. Now, it is natural to ask for semicomplete split digraphs, and they proposed the following conjecture.

Conjecture 1.6. *[9] Every 2-arc-strong semicomplete split digraph D contains a good (u, v)-pair for every choice of vertices u, v in D.*

In this paper, we prove that every 2-arc-strong semicomplete split digraph D contains a good (u, v)-pair for every choice of vertices u, v of D, which confirms Conjecture 1.6.

Theorem 1.7. *Every 2-arc-strong semicomplete split digraph D contains a good (u, v)-pair for every choice of vertices u, v of D.*

2 The Proof of the Main Result

Based on Theorem 1.5, we need to consider the semicomplete split digraphs with the structure illustrated in **(1)**, **(2)** and the semicomplete split digraphs in Appendix of [2].

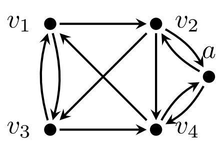
We first consider the semicomplete split digraphs in Appendix of [2], and then discuss the semicomplete split digraphs with the structure illustrated in **(1)** and **(2)** of Theorem 1.5.

In the first part, the main idea is to consider the common subdigraphs of these cases. By finding special good pairs in S_4 and then transforming them into good pairs in $(\bar{i})-(\overline{v})$, we can verify that there is a good (u,v)-pair for every choice of u,v in $(\bar{i})-(\overline{v})$. Then based on the results on $(\bar{i})-(\overline{v})$, we can verify that there is a good (u,v)-pair for every choice of u,v in the combinations of the five basic cases by checking the pairs (a,b) and (b,a). In the second part, we verify that all the semicomplete split digraph with structure in **(1)** have been covered by those in the Appendix, and there exists no semicomplete split digraph with structure in **(2)**.

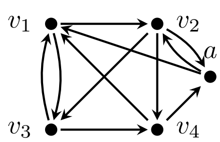
For the rest of this paper, we denote an out-branching (resp., in-branching) rooted at t by t-out-branching (resp., t-in-branching) for simplicity.

2.1 Five Basic Cases in Appendix

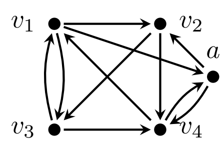
The following figures show the structures of the five basic cases. We say they are 'basic' since any other cases on 6 vertices can be somehow viewed as the combinations of these 5 cases.



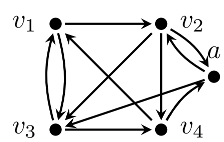
$(\bar{i})$



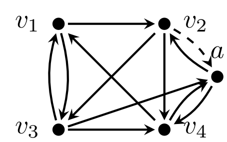
$(\overline{ii})$



$(\overline{iii})$

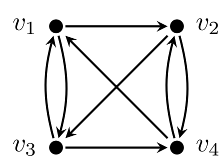


$(\overline{iv})$



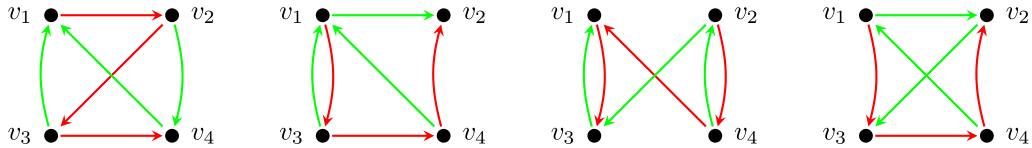
$(\overline{v})$

We need the following lemma.



S_4

Lemma 2.1. *Each vertex pair in S_4 is a good vertex pair.*



(v_1, v_1) in S_4 (v_1, v_2) in S_4 (v_2, v_1) in S_4 (v_1, v_3) in S_4

Proof. Since S_4 is vertex-transitive, we only need to check the following four vertex pairs.

(v_1, v_1) v_1-out-branching: $\{v_1v_2, v_2v_3, v_3v_4\}$; v_1-in-branching: $\{v_2v_4, v_4v_1, v_3v_1\}$.

(v_1, v_2) v_1-out-branching: $\{v_1v_3, v_3v_4, v_4v_2\}$; v_2-in-branching: $\{v_3v_1, v_4v_1, v_1v_2\}$.

(v_2, v_1) v_2-out-branching: $\{v_2v_4, v_4v_1, v_1v_3\}$; v_1-in-branching: $\{v_4v_2, v_2v_3, v_3v_1\}$.

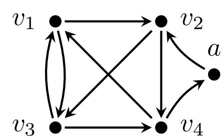
(v_1, v_3) v_1-out-branching: $\{v_1v_3, v_3v_4, v_4v_2\}$; v_3-in-branching: $\{v_4v_1, v_1v_2, v_2v_3\}$.

So all the vertex pairs in S_4 are good vertex pairs by symmetry. □

Now we are ready to prove that each vertex pair in the five basic cases is a good vertex pair.

Theorem 2.2. *Each vertex pair in the five basic cases is a good vertex pair.*

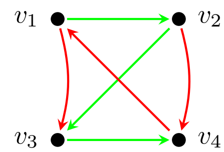
Proof. Suppose that D is one of the basic cases above. Observe that all the five basic cases have a common subdigraph D_1. By replacing the 2-path v_4av_2 with the arc v_4v_2 in D_1, we get S_4 as the resulting graph. And from Lemma 2.1, we know that any vertex pair in S_4 is a good vertex pair.



(D_1)

We first claim that any vertex pair (v_j, v_i) in D is a good vertex pair for $i, j \in [4]$. By the above discussion, we can find arc-disjoint w-in-branching W and t-out-branching T for any vertex pair (t, w) in S_4. If both W and T do not contain v_4v_2, then we add av_2 and v_4a to W and T respectively, to obtain W' and T'. It is not hard to see that W' is a w-in-branching and T' is a t-out-branching in D_1, and they are arc-disjoint. If any vertex pair (v_j, v_i) in D_1 is a good vertex pair, certainly it is also a good vertex pair in D since D_1 is a subdigraph of D. If W contains the arc v_4v_2, then we replace v_4v_2 by v_4a and av_2 to obtain W'. Since a has another in-neighbor v_n in D, we can add v_na to T to obtain T'. One can check the fact that T' is a t-out-branching in D, W' is a w-in-branching in D, and they are arc-disjoint. Similarly, we can discuss the case when the t-out-branching contains the arc v_4v_2.

The vertex pair (a, a) is indeed a good vertex pair because we can add av_2 and v_4a, respectively, to the following v_2-out-branching and v_4-in-branching in S_4.



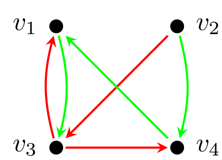
(v_2, v_4) in S_4

Now we consider the vertex pairs (a, v_i) in D, for $i \in [4]$. Suppose that two distinct out-neighbors of a are v_2 and v_m in D. If we can find a good vertex pair of one of the following two types in S_4, then by a similar discussion above, we can transform it to a good vertex pair (a, v_i) in D.

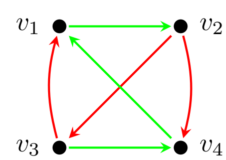
type 1: (v_2, v_i) is a good vertex pair in S_4 and one of the v_i-in-branching does not contain v_4v_2;

type 2: (v_m, v_i) is a good vertex pair in S_4 and one of the v_i-in-branching contains v_4v_2;

We first show how to transform such a vertex pair to a good vertex pair in D, and then verify every v_i is in such a vertex pair for $i \in [4]$. If we have a vertex pair of type 1, then we can find a v_i-in-branching W which does not contain v_4v_2 and a v_2-out-branching T in S_4. We add av_2 and av_m respectively to W and T, and this gives us arc-disjoint v_i-in-branching and a-out-branching in D. Similarly, if we have a vertex pair of type 2, we replace v_4v_2 by v_4a and av_2 in the v_i-in-branching and add av_m to the v_m-out-branching in S_4. This also gives us arc-disjoint v_i-in-branching and a-out-branching in D. So it suffices to show any v_i is in a vertex pair of type 1 or type 2. Note that (v_2, v_4) is a vertex pair of type 1 since any v_4-in-branching could not contain v_4v_2 in S_4. And (v_2, v_3) and (v_2, v_2) are also vertex pairs of type 1 since we have the following structures in S_4:

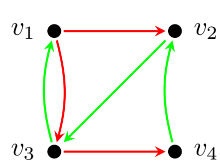


(v_2, v_3) in S_4

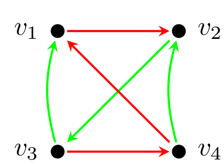


(v_2, v_2) in S_4

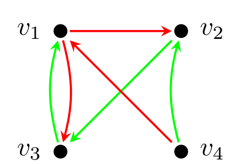
We now show (a, v_1) is a good vertex pair in D, and it suffices to show (v_m, v_1) is a vertex pair of type 2. Note that we have $v_m \in \{v_1, v_3, v_4\}$ in the five basic cases, so we only need to check whether (v_1, v_1), (v_3, v_1) and (v_4, v_1) are all of type 2. In fact, we have the following structures in S_4, which means that the three vertex pairs are of type 2:



(v_1, v_1) in S_4



(v_3, v_1) in S_4



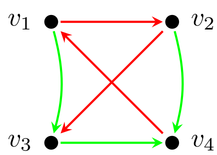
(v_4, v_1) in S_4

As for the vertex pairs (v_i, a) in D, in which $i \in [4]$, we suppose that two distinct in-neighbors of a are v_4 and v_n in D. Similarly, we consider the following two types of vertex pairs:

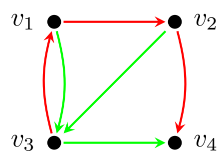
type $1'$: (v_i, v_4) is a good vertex pair in S_4 and one of the v_i-out-branching does not contain v_4v_2;

type $2'$: (v_i, v_n) is a good vertex pair in S_4 and one of the v_i-out-branching contains v_4v_2.

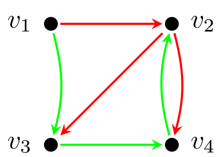
We list the required structures below and omit the discussions here since it is very similar to those for vertex pairs (a, v_i) in D.



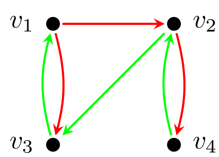
(v_4, v_4) in S_4



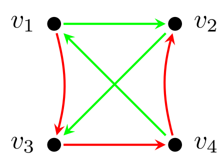
(v_3, v_4) in S_4



(v_1, v_2) in S_4



(v_1, v_1) in S_4

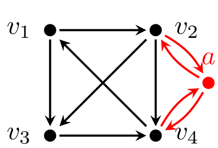


(v_1, v_3) in S_4

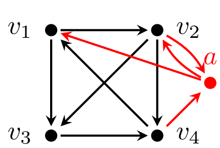
□

2.2 Combinations of the Five Basic Cases in Appendix

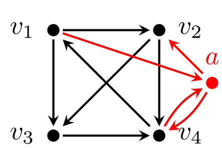
By deleting v_3v_1 in the five basic cases, we get the following structures $(i) - (v)$.



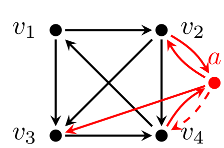
(i)



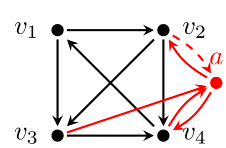
(ii)



(iii)

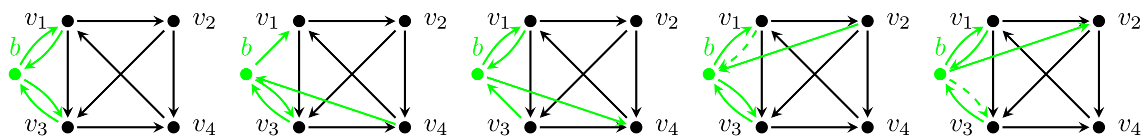


(iv)



(v)

Reverse all the arcs in $(i) - (v)$, rotate 180 degrees clockwise, and relabel, we can obtain the corresponding reversed and rotated structures $(i)^* - (v)^*$.



$(i)^*$ $(ii)^*$ $(iii)^*$ $(iv)^*$ $(v)^*$

We say D is a combination of (e) and $(f)^*$, in which $e, f \in \{i, ii, iii, iv, v\}$, if D is the digraph with $V(D) = \{a, b, v_1, v_2, v_3, v_4\}$ and $A(D) = A((e)) \cup A((f)^*)$, and we denote it by $(e) \times (f)^*$.

Note that all left cases in Appendix of [2] can be represented as some $(e) \times (f)^*$, we only need to prove the following theorem to complete the proof for all cases in Appendix of [2].

Theorem 2.3. *Each vertex pair in $(e) \times (f)^*$ is a good vertex pair, in which $e, f \in \{i, ii, iii, iv, v\}$.*

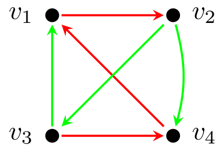
Proof. Suppose that D is the combination of (e) and $(f)^*$.

We first replace the 2-path v_3bv_1 with v_3v_1 and remove all other arcs that are incident to b in D to obtain a new graph D'. By Theorem 2.2, we can find arc-disjoint w-in-branching W and t-out-branching T for any vertex pair (t, w) in D'. If W contains the arc v_3v_1, then we replace v_3v_1 by v_3b and bv_1 to obtain W', observe that W' is a w-in-branching in D. Since b has another in-neighbor v_n in D, we add v_nb to T to obtain T', and obviously T' is a t-out-branching in D and it is arc-disjoint with W'. Similarly, we can discuss the case when t-out-branching contains the arc v_3v_1. So we may assume that W and T do not contain v_3v_1. In this case, we can simply add bv_1 and v_3b to W and T respectively to obtain a w-in-branching and a t-out-branching in D, and they are arc-disjoint. This means that any vertex pair (t, w) in which $w, t \in \{v_1, v_2, v_3, v_4, a\}$ is a good vertex pair in D.

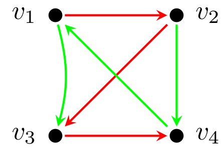
Similarly, we replace the 2-path v_4av_2, remove all other arcs that are incident to a in D, and reverse all the arcs to obtain a new graph D'. Note that reversing all the arcs would only change a good vertex pair (t, w) in the original graph into a good vertex pair (w, t) in the resulting graph, so by analogous discussion as above, we deduce that any vertex pair (t, w) in which $w, t \in \{v_1, v_2, v_3, v_4, b\}$ is a good vertex pair in D.

Now the only vertex pairs we need to consider in the following part are (a, b) and (b, a). If (a, b) (resp., (b, a)) is a good vertex pair in $(e) \times (f)^*$, then (a, b) (resp., (b, a)) is also a good vertex pair in $(f) \times (e)^*$ since $(e) \times (f)^*$ can be transformed into $(f) \times (e)^*$ by reversing all the arcs and relabeling, where e and f are elements of $\{i, ii, iii, iv, v\}$.

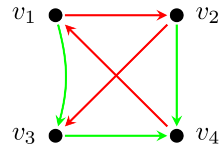
We suppose a (resp., b) has two out-neighbors v_2, v_{m_1} (resp.,v_1,v_{m_2}) and two in-neighbors v_4, v_{n_1} (resp., v_3,v_{n_2}). We can find the following 4 structures in S_4, in which each can be transformed into the desired in-branching and out-branching in D. We use Structure 1 to show how to achieve this. Note that av_2 is in every D, so if v_2b is in D (namely, D contains $(iv)^*$), we can add av_2 to the v_2-out-branching and replace v_3v_1 by v_3b, bv_1 to obtain an a-out-branching in D. Then by adding v_2b, av_{m_1} to the v_2-in-branching, we can get a b-in-branching in D. So we can always find arc-disjoint b-in-branching and a-out-branching in any D that contains $(iv)^*$ or (iv). Namely, (a, b) would be a good vertex pair in such D. Similarly, we have: Structure 2 implies (a, b) is a good vertex pair in D that contains (ii) and (b, a) is a good vertex pair in D that contains (v), and Structure 3 implies (b, a) is a good vertex pair in D that contains (iii). Moreover, we can find Structure 4 in any D that contains (i), which implies (b, a) is a good vertex pair in such D.



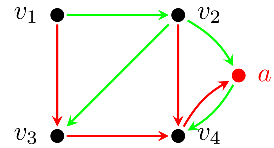
Structure 1
$(iv)\,(a, b)$



Structure 2
$(ii)\,(a, b)$ & $(v)\,(b, a)$



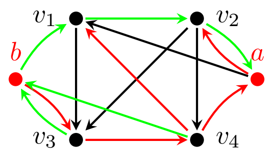
Structure 3
$(iii)^*\,(b, a)$



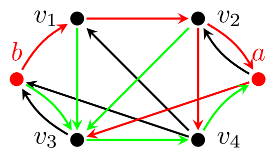
Structure 4
$(i)\,(b, a)$

So we only need to check the following vertex pairs (total 9 distinct vertex pairs): (b, a) of $(e) \times (f)^*$ in which $e, f \in \{ii, iv\}$ and (a, b) of $(e) \times (f)^*$ in which $e, f \in \{i, iii, v\}$.

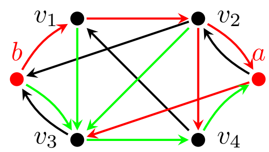
For (b, a), we have the following results:



$(ii)^* \times (ii)$

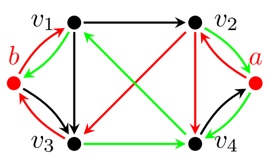


$(ii)^* \times (iv)$

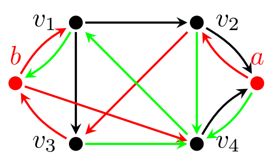


$(iv)^* \times (iv)$

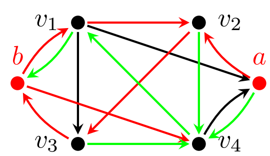
For (a, b), we have the following results:



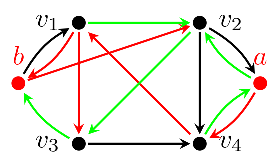
$(i)^* \times (i)$



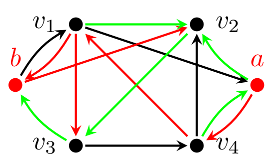
$(iii)^* \times (i)$



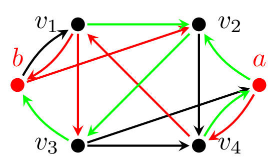
$(iii)^* \times (iii)$



$(v)^* \times (i)$



$(v)^* \times (iii)$



$(v)^* \times (v)$

We are done now.

□

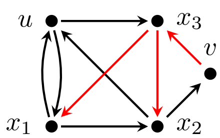
2.3 Semicomplete Split Digraph with Specific Structure

If D has structures illustrated in Theorem 1.5**(2)**, then observe that $u \in V_1, c \in V_2$, and $uc, cu \notin A$, we have D is not a semicomplete split digraph.

If D has structures illustrated in Theorem 1.5**(1)**, and D is a semicomplete split digraph, then:

1. When $u \in V_1$: Since all arcs incident to u have been illustrated and D is a semicomplete split digraph, we have $V_2 = \{x_1, x_2, x_3\}$. If there is another vertex $t \in V_1 \setminus \{u\}$, then since D is 2-arc-strong, t has at least 2 in-arcs, which implies that t can only be v.

   - If $V_1 = \{u\}$, then since D is a 2-arc-strong semicomplete digraph, it means either D has a strong arc decomposition or D is S_4.
   - If $V_1 = \{u, v\}$, then $x_3x_1, x_2x_1 \in A(D)$ since $N_D^+(x_1) = \{x_2, u\}, N_D^+(x_2) = \{v, u\}$. Since x_3 has at least 2 in-arcs, we have $vx_3 \in A(D)$. This structure has been discussed in the Appendix of [2], the result of which is either D has a strong arc decomposition or D is a counterexample listed in Appendix, and we have proved that D always has good (u, v)-pairs for all $u, v \in D$.



2. When $u \in V_2$: Since u is only adjacent to x_1, x_2, x_3, and the subdigraph induced by V_2 is semicomplete, then we have $|V_2| \leq 4$. Every vertex in V_1 is adjacent to u since D is a semicomplete split digraph and so we have $V_1 \cup V_2 = \{u, x_1, x_2, x_3\}$.

   - If $|V_2| = 1$, then $x_1x_2 \in D$ contradicts the fact that V_1 is an independent set.
   - If $|V_2| = 2$, then at least two of x_1, x_2, x_3 belong to V_1. Since D is 2-arc-strong, each vertex x in V_1 has at least 2 in-arcs and 2 out-arcs, which means x has a 2-cycle with u. And so there would be two 2-cycles incident to u, which contradicts the structures as illustrated.
   - If $|V_2| = 3, 4$, then D is a 2-arc-strong semicomplete digraph, which means either D has a strong arc decomposition or D is S_4.

So, we are done.

3 Concluding Remarks

In this paper, we have considered good pairs in semicomplete split digraphs. Based on the results from [2], we only need to examine a finite number of graphs. In addition to the proofs above, we also verified all the possibilities with the assistance of a computer, and we have included the code in the Appendix.

After proving that every 2-arc-strong semicomplete split digraph contains a good (u, v)-pair for any choice of vertices $u, v \in D$, it is natural to turn our attention to 2-arc-strong split digraphs in general. Bang-Jensen and Wang [5] already showed that 2-arc-strong is not sufficient to guarantee the existence of a good (u, v)-pair for any choice of vertices u, v in a split digraph. In fact, all of their counterexamples belong to the class described in **(1)** of Theorem 1.5. In [2], Ai et al. presented the following corollary:

Corollary 3.1. *For any 2-arc-strong split digraph $D = (V_1, V_2; A)$, it has a strong arc decomposition by adding at most 1 specific arc in $D[V_2]$.*

Using this corollary, it may be possible to fully characterize 2-arc-strong split digraphs that contain a good (u, v)-pair for any choice of vertices u, v.

References

[1] J. Ai, S. Gerke, G. Gutin, S. Wang, A. Yeo, and Y. Zhou. On Seymour's and Sullivan's second neighbourhood conjectures. *J. Graph Theory.*, 105 (2024), no. 3, 413–426.

[2] J. Ai, F. He, Z. Li, Z. Qin, and C. Wang. A complete characterization of split digraphs with a strong arc decomposition. *arXiv preprint*, arXiv:2408.02260, 2024. URL: `https://arxiv.org/abs/2408.02260`.

[3] J. Bang-Jensen, S. Bessy, F. Havet, and Y. Anders. Arc-disjoint in- and out-branchings in digraphs of independence number at most 2. *J. Graph Theory*, 100 (2022), no. 2, 294–314.

[4] J. Bang-Jensen and G. Gutin. Digraphs-Theory, Algorithms and Applications, 2nd Ed.. Springer Monographs in Mathematics, London: Springer-Verlag London, Ltd., 2009.

[5] J. Bang-Jensen, G. Gutin and A. Yeo. Arc-disjoint strong spanning subdigraphs of semicomplete compositions. *J. Graph Theory*, 95(2):267–289, 2020.

[6] J. Bang-Jensen and J. Huang. Decomposing locally semicomplete digraphs into strong spanning subdigraphs. *J. Combin. Theory Ser. B*, 102:701–714, 2010.

[7] J. Bang-Jensen and Y. Wang. Arc-disjoint out- and in-branchings in compositions of digraphs. *J. European J. Combin*, 120 (2024), Paper No. 103981, 28 pp.

[8] J. Bang-Jensen and Y. Wang. Arc-disjoint out-branchings and in-branchings in semicomplete digraphs. *J. Graph Theory*, 106 (2024), no. 1, 182–197.

[9] J. Bang-Jensen and Y. Wang. Strong arc decomposition of split digraphs. *J. Graph Theory*, 2024, https://doi.org/10.1002/jgt.23157.

[10] J. Bang-Jensen and A. Yeo. Decomposition k-arc-strong tournaments into strong spanning subdigraphs. *Combinatorica*, 24(3):331–349, 2004.

[11] C. Thomassen. Configurations in graphs of large minimum degree, connectivity, or chromatic number. *Annals of the New York Academy of Sciences*, 555:402–412, 1989.

4 Appendix

```
install.packages("igraph")
library(igraph)
set.seed(124)

# main functions
check_in_out_branching= function(edge_set,D,sample_size, details=F){
  vertex_set = unique(edge_set)
  l = length(vertex_set)
  check_matrix = matrix(0,nrow = l,ncol = l)
  colnames(check_matrix) = vertex_set
  rownames(check_matrix) = vertex_set
  tmp <- tempfile(pattern="image",tmpdir=".",fileext = ".jpg")
  jpeg(tmp, width=1000*l, height=1000*l)
  par(mfrow = c(l,l),cex = 3)
  for(i in 1:l){
    for(j in 1:l){
        for(k in 1:sample_size){
          t_ = subgraph.edges(D,sample_spanning_tree(D))
          t_check = dominator_tree(t_,vertex_set[i],mode = "in")
          if(length(t_check$leftout)==0){
            D_ = difference(D,t_)
            t_out = dominator_tree(D_,vertex_set[j],mode = "out")
            if(length(t_out$leftout)==0){
              check_matrix[i,j]=1
              if(details==T){
                print(plot(t_check$domtree,main =paste0(vertex_set[i],"
                    -in-branching")))
                print(plot(t_out$domtree,main =paste0(vertex_set[j],"-
                    out-branching")))
              }
              break
            }
          }
        }
    }
  }
  dev.off()
  return(check_matrix)}     #details(default: False): whether to create
      a new .JPG file to show all the in-branchings and out-branchings

judge_in_out = function(check_matrix){
  a = nrow(check_matrix)
  all(check_matrix)!=0)
```

```
}


#input all the cases that should be checked
common_set = c("v1","v2","v2","v4","v3","v4","v1","v3","v4","v1","v2","v3")
a1_type1 = c("v2","a1","a1","v2","a1","v4","v4","a1")
a1_type2 = c("v2","a1","a1","v2","a1","v1","v4","a1")
a1_type3 = c("a1","v2","v1","a1","a1","v4","v4","a1")
a1_type4 = c("v2","a1","a1","v2","a1","v3","v4","a1")
a1_type5 = c("a1","v2","v3","a1","a1","v4","v4","a1")

a2_type1 = c("a2","v1","v1","a2","a2","v3","v3","a2")
a2_type2 = c("a2","v1","v4","a2","v3","a2","a2","v3")
a2_type3 = c("v1","a2","a2","v1","a2","v4","v3","a2")
a2_type4 = c("a2","v1","v2","a2","a2","v3","v3","a2")
a2_type5 = c("a2","v1","v1","a2","a2","v2","v3","a2")

edge_set1 = c(common_set,"v3","v1",a1_type1)

edge_set2 = c(common_set,"v3","v1",a1_type2)

edge_set3 = c(common_set,"v3","v1",a1_type3)

edge_set4 = c(common_set,"v3","v1",a1_type4)

edge_set5 = c(common_set,"v3","v1",a1_type5)

edge_set6 = c(common_set,a1_type1,a2_type1)

edge_set7 = c(common_set,a1_type1,a2_type2)

edge_set8 = c(common_set,a1_type1,a2_type3)

edge_set9 = c(common_set,a1_type1,a2_type4)

edge_set10 = c(common_set,a1_type2,a2_type2)

edge_set11 = c(common_set,a1_type2,a2_type3)

edge_set12 = c(common_set,a1_type2,a2_type4)

edge_set13 = c(common_set,a1_type3,a2_type3)

edge_set14 = c(common_set,a1_type3,a2_type4)

```

```
edge_set15 = c(common_set,a1_type3,a2_type5)

edge_set16 = c(common_set,a1_type4,a2_type4)



for(i in 1:16){
  edge_set = get(paste0("edge_set",as.character(i)))
  D = graph(edges = edge_set ,directed = T)
  check_matrix = check_in_out_branching(edge_set,D,2000)
  print(check_matrix)
  print(judge_in_out(check_matrix))
}
```