Scientometrics
https://doi.org/10.1007/s11192-022-04543-y

Does main path analysis prefer longer paths?

Chung-Huei Kuan[1,2]

Received: 18 July 2021 / Accepted: 30 September 2022

Abstract
Does main path analysis (MPA), in producing the main paths (MPs), invariably choose the longer paths over the shorter ones? This work examines the various combinations of the most popular path search algorithms, i.e., local and global searches (including the key-route variant), and the weight assignment algorithms, i.e., search path count (SPC), search path link count (SPLC), and search path node pair (SPNP), to investigate the path preference of MPA. Based on a simplified model, this work finds that, when there are multiple paths between a pair of nodes, MPA indeed goes for the longer paths under some, but not all, algorithm combinations, but it may retain both longer and shorter paths under other combinations.

Keywords Main path analysis · Main path · Weight assignment · Path search · Longer path

Introduction

It is quite extraordinary that Main Path Analysis (MPA) (Hummon & Doreian, 1989), a conceptually straightforward network analytical method, would receive such wide acceptance among researchers. A rough search using Google Scholar with the key term "main path analysis" reveals that, only after 2021, there are already more than 430 publications.

One of the factors contributing to MPA's popularity is that it may reduce a large and complex network, usually embodying the papers or patents of a research or technological field and their citations, into one or more main paths (MPs) consisting of only a handful of connected nodes and arcs, claimed to be epitomizing evolution trajectories of the field. For example, Park and Magee (2017) indicated that "a major reason to use main path analysis is to reduce the network complexity". Another factor driving MPA's popularity is its immediate availability from the acclaimed network analysis software Pajek (Batagelj & Mrvar, 1998; De Nooy et al., 2018).

✉ Chung-Huei Kuan
maxkuan@mail.ntust.edu.tw

1 Graduate Institute of Patent, National Taiwan University of Science and Technology, No. 43 Sec. 4 Keelung Rd., Taipei 10607, Taiwan, ROC

2 Center for Research in Econometric Theory and Applications, National Taiwan University, No. 1, Sec. 4, Roosevelt Rd., Taipei 10617, Taiwan, ROC

Published online: 17 October 2022

The traditional MPA is restricted to a directed and acyclic network, and requires two major steps. Firstly, each arc of the network is assigned a weight based on a traversal count of the arc. Then, based on the weights assigned to the arcs, the MPs are sought as series of end-to-end connected arcs from source to sink nodes. There are various weight assignment and path search algorithms and those most popular ones will be detailed in the subsequent section Revisiting MPA.

As a popular method, there are invariably various attempts to improve MPA by addressing some of its inherent limitations. Due to the numerousness of these improvement works, the following are a short list of samples. For examples, focusing on the MPA's weight assignment, Jiang et al. (2020) proposed a SimSPC algorithm for extracting MPs from a cyclic citation network, thereby overcoming one of MPA's requirement that the analyzed citation network has to be acyclic. Yu and Pan (2021), instead of treating each arc (citation) equally, incorporated discipline difference and time span between the cited and the citing into the original arc weights. Similarly, Chen et al. (2022) combined the semantic similarity between the cited-citing pair into the conventional weights so that the derived MPs would reveal better "topic coherence". In addition to the improvements related to the weight assignment, there are also different proposals for new ways to search the MPs. For examples, Yoon et al. (2020) first divided a technology domain into hierarchically structured sub-domains, generated main paths for each sub domain, and integrated all main paths together into hierarchical main paths. To uncover multiple main streams of a field, Kim et al. (2022), rather than using the traditional path search algorithms, discovered multiple longest paths from the citation network and merged them together.

This study is inspired by some recent works proposing ways of improving MPA, where the researchers, in specifying their motivation and criticizing some deficiency of traditional MPA, claimed that the traditional MPA "tends to favour longer paths over shorter ones" and this is "an undesirable property that leads to counterintuitive results" (Šubelj et al., 2020). Some researchers also followed the same argument (cf. Xu et al., 2022). Whether a longer MP is "counterintuitive" is debatable (some discussion is left to the "Summary" section), what is intriguing is: does MPA always pick the paths of longer lengths when there are other candidates for selection? Considering the popularity of MPA, it is interesting to notice that this question, to the knowledge of the author of the present study, is not yet answered. Some researchers did mention their intuitive observations about the greediness of MPA (cf. Lu et al., 2021). The author of the present study himself asserted in a prior study that MPA seems to favor longer paths under a specific path search algorithm (Kuan et al., 2019). However, none has provided a formal investigation and explained whether this intuition holds water under different variants of MPA.

Therefore, to fill in the gap, the study conducts a thorough and systematic investigation on the path preference of MPA under the most popular path search and weight assignment algorithms.

Revisiting MPA

As mentioned in the Introduction, the gist of the MPA lies in weight assignment and path search. There are various weight assignment algorithms, such as *search path count* (SPC) (Batagelj, 2003), *search path link count* (SPLC), *search path node pair* (SPNP), *node pair projection count* (NPPC) (all from Hummon and Doreian (1989)), *forward citation node pair* (FCUR) (Choi & Park, 2009), etc. SPC, SPLC, SPNP, as they are available from

Pajek, are the most popular ones. All three algorithms assign the number of times an arc is traversed between two sets of nodes as its weight. SPC, SPLC, and SPNP differ only in the sets of nodes they consider in calculating the traversal counts.

There are also various path search algorithms. The most popular ones, again due to their availability from Pajek, are *global search* (Liu & Lu, 2012) and *local search* (Hummon & Doreian, 1989), which all identify one or more MPs extending between the network's sources (i.e., nodes with only outgoing arcs) and sinks (i.e., nodes having only incident arcs).

The global search algorithm has two variants. The *global standard* variant, for the paths between all pair of sources and sinks, sums the weights of their constituent arcs and selects the ones having the greatest accumulated weight as the MPs. The other variant, *global key-route*, will be described later.

The local search algorithm has three variants. The *local forward* variant starts from the sources, traces the highest weighted outgoing arcs forward to their terminal nodes, and repeats this forward-tracing-through-highest-weighted-arcs process stage by stage until a sink is reached. The sets of arcs thus traced are identified as the MPs. The *local backward* variant works just like its forward counterpart, except that it starts from the sinks and backtracks the highest weighted incoming arcs to their start nodes. When conducting local search using Pajek, an analyst may set a tolerance (e.g. 10%) so that, not only the highest weighted arcs are picked, but also those whose weights are smaller than yet still within the tolerance of the highest weight (e.g., arcs weighing between 90 and 100% of the highest weight for a 10% tolerance).

As to the *global key-route* and *local key-route* variants (Liu & Lu, 2012), one or more highest weighted arcs, called *key routes*, have to be specified first. If the arcs of the highest weight are selected, it is called global or local key-route 1. If those of the highest or the second highest weights are chosen, it is called global or local key-route 2. In other words, key-route n means the search starts with arcs having one of the topmost n weights and there may be n or more such key routes. Then, the global key-route variant respectively finds the paths having the greatest accumulated weights between the sources and the start nodes of the key routes (i.e., paths preceding the key routes), and the paths between the terminal nodes of the key routes and the sinks (i.e., paths succeeding the key routes). These preceding and succeeding paths, together with their intermediate key routes, constitute the MPs. The local key-route variant, on the other hand, determines the paths preceding and succeeding the key routes respectively in the local backward manner from the start nodes, and in the local forward manner from the terminal nodes, of the key routes.

A simplified model

Depending on the weight assignment and path search algorithms used, MPA may produce a single MP or multiple MPs from the network. For the single MP or each of the MPs found by MPA under a specific combination of the weight assignment and path search algorithms, does MPA always pick the paths of longer lengths when there are other candidates for selection?

To answer this question, firstly, it is more specifically rephrased as, for a pair of nodes with multiple paths in between, whether MPA always picks the paths consisting of the higher numbers of arcs.

Then, a simplified model that may reflect MPA's preference between the pair of nodes is employed and shown in Fig. 1, where two sub-nets are connected through their respective nodes s and t. There are two paths between s and t: a shorter one consisting of a single arc (s, t) and a longer one involving an intermediate node d and consisting of two series-connected arcs (s, d) and (d, t).

Of course, from node s to node t, there may be other possible connections. The simplified model reduces all possible connections to only a longer path $s \rightarrow d \rightarrow t$ and a shorter one $s \rightarrow t$, as MPA's preference between them reveal what MPA would behave under more complex connection scenarios. For example, by adding two more nodes m, n and some arcs between nodes s and t, there are now four paths that goes from node s to node t: $s \rightarrow t$, $s \rightarrow d \rightarrow t$, $s \rightarrow m \rightarrow d \rightarrow t$ and $s \rightarrow m \rightarrow \mathrm{n} \rightarrow t$. If MPA is indeed greedy and chooses $s \rightarrow d \rightarrow t$ in the simplified model, it will then go for the longer path $s \rightarrow m \rightarrow d \rightarrow t$ or $s \rightarrow m \rightarrow \mathrm{n} \rightarrow t$ or both in this more complex scenario. Otherwise, MPA will choose or retain the shorter path $s \rightarrow t$ or $s \rightarrow d \rightarrow t$ if MPA makes the same decision that follows $s \rightarrow t$ in the simplified model.

More specifically, if at least one of the MPs derived by MPA would extend from sub-net 1 to sub-net 2, the MP then must run through the nodes s and t. For this MP, if MPA always picks the longer path involving more arcs, the MP will detour through the intermediate node d. On the other hand, if MPA goes for the shorter path or keeps both paths as part of the MP, MPA cannot be said to always favor longer paths.

A closer look at weight assignment

To see whether MPA will pick the longer path involving (s, d) and (d, t) or the shorter one (s, t) in Fig. 1, these arcs' respective weights under SPC, SPLC, and SPNP must be decided first.

As mentioned in Introduction, SPC, SPLC, and SPNP all calculate the number of times an arc is crossed between two sets of nodes, and they differ only in the sets of nodes they involve.

There are four sets of nodes that are respectively considered by SPC, SPLC, and SPNP. Using the arc (s, t) of Fig. 1 as an example, they are the sources that may cross the arc (s, t) to reach the terminal node t (denoted as *source*(s, t) and depicted as black nodes in sub-net

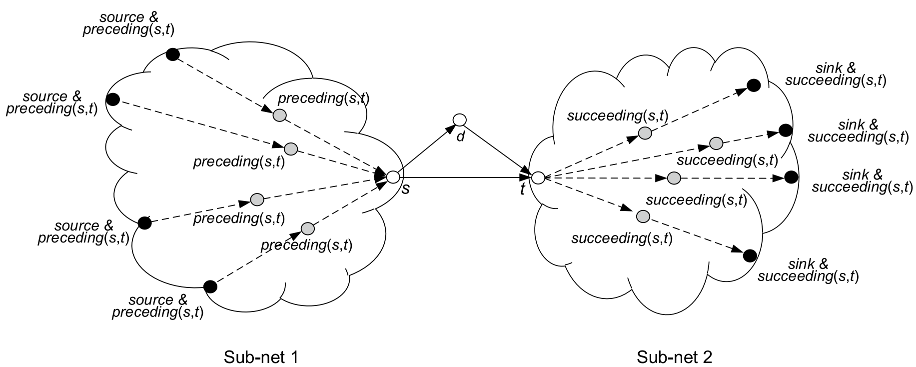

Fig. 1 A simplified model for investigating the preference of MPA

1), the sinks that may be reached from the start node s after crossing (s, t) (denoted as *sink*(s, t) and depicted as black nodes in sub-net 2), all nodes preceding (s, t) that may cross (s, t) to reach t (denoted as *preceding* (s, t) and including *source* (s, t), the intermediate grey nodes, and s in sub-net 1), and all succeeding nodes that may be reached from s after crossing (s, t) (denoted as *succeeding* (s, t) and including *sink* (s, t), the intermediate grey nodes, and t in sub-net 2).

Then, the SPC, SPLC, and SPNP weights of (s, t), denoted as W_{SPC} (s, t), W_{SPLC} (s, t), and W_{SPNP} (s, t), are respectively the number of times (s, t) is crossed from *source* (s, t) to *sink* (s, t), from *preceding* (s, t) to *sink* (s, t), and from *preceding* (s, t) to *succeeding* (s, t). Formally, W_{SPC} (s, t), W_{SPLC} (s, t), and W_{SPNP} (s, t) are specified in Eqs. (1)–(3).

$$W_{\text{SPC}}(s,t) = \left(\sum\nolimits_{\forall i \in \, source\,(s,t)} C(i,t)\right)\left(\sum\nolimits_{\forall j \in sink(s,t)} C(s,j)\right), \tag{1}$$

$$W_{\text{SPLC}}(s,t) = \left(\sum\nolimits_{\forall i \in \, preceeding\,(s,t)} C(i,t)\right)\left(\sum\nolimits_{\forall j \in sink(s,t)} C(s,j)\right), \text{ and} \tag{2}$$

$$W_{\text{SPNP}}(s,t) = \left(\sum\nolimits_{\forall i \in \, preceeding\,(s,t)} C(i,t)\right)\left(\sum\nolimits_{\forall j \in succeeding(s,t)} C(s,j)\right), \tag{3}$$

where $C(u,v) = \begin{cases} 0, \text{if } u = v, \text{ or} \\ \text{number of paths from } u \text{ to } v, \text{if } u \neq v \end{cases}.$

Therefore, $\sum_{\forall i \in source(s,t)} C(i,t)$ is the number of paths from *source* (s, t) to t, $\sum_{\forall i \in preceeding(s,t)} C(i,t)$ is the number of paths from *preceding* (s, t) to t, $\sum_{\forall j \in sink(s,t)} C(s,j)$ is the number of paths from s to *sink* (s, t), and $\sum_{\forall j \in succeeding(s,t)} C(s,j)$ is the number of paths from s to *succeeding* (s, t).

Since $source(s,t) \subseteq preceeding(s,t)$ and $sink(s,t) \subseteq succeeding(s,t)$, it is clearly that W_{SPC} (s, t) ≤ W_{SPLC} (s, t) ≤ W_{SPNP} (s, t).

To demonstrate Eqs. (1) to (3), Fig. 2 depicts a fictitious network whose SPC, SPLC, and SPNP arc weights are shown beneath the arcs in the format SPC/SPLC/SPNP. Using the highest-weighted arc (6, 8) as an example, *source* (6, 8) = {1, 2, 3}, *sink* (6, 8) = {14, 15, 16, 17, 18}, *preceding* (6, 8) = {1, 2, 3, 6}, and *succeeding* (6, 8) = {8, 11, 12 14, 15, 16, 17, 18}. Then, W_{SPC} (6, 8) = 21 = 3·7 as there is one path from each of the three sources 1–3 to the node 8 (3 = 1·3) and, from the node 6, there are one path to each of the three sinks 14–16 and two paths to the two sinks 17 and 18 (7 = 1·3 + 2·2). W_{SPLC} (6, 8) = 28 = 4·7 as there is one path from each of the four preceding nodes 1–3 and 6 to the node 8 (4 = 1·4), and there are the same 7 paths from the node 6 to *sink* (6, 8). W_{SPNP} (6, 8) = 44 = 4·11 as there are the same 4 paths from *preceding* (6, 8) to the node 8 and, from the node 6, there are one path to each of the five succeeding nodes 8, 11 and 14–16, and two paths to each of the three succeeding nodes 12, 17, 18 (11 = 1·5 + 2·3). Global standard and key-route 1, local forward and backward (with zero tolerance), and local key-route 1 algorithms all produce the same MP denoted by the black nodes and arcs.

MPA preference under various algorithms

For the network shown in Fig. 1, if there are

n_1 paths from *source* (s, t) to t,

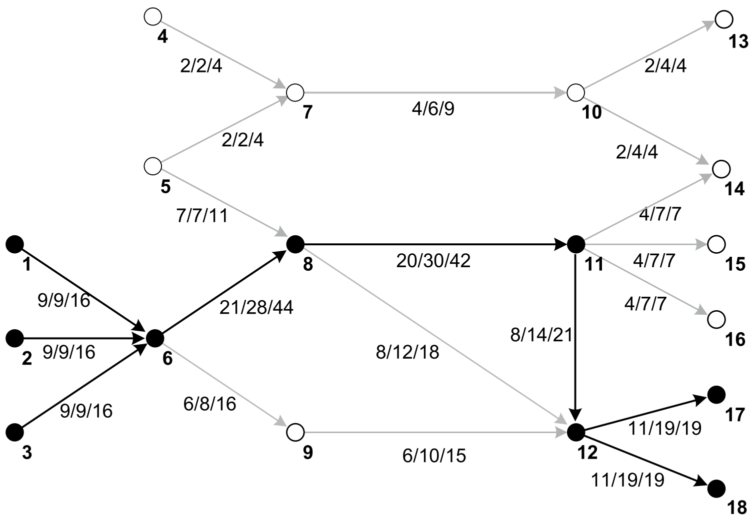

Fig. 2 A fictitious network with SPC/SPLC/SPNP arc weights

n_2 paths from *preceding* (s, t) to t $(n_1 \le n_2)$,
m_1 paths from s to *sink* (s, t), and
m_2 paths from s to *succeeding* (s, t) $(m_1 \le m_2)$,
the SPC, SPLC, and SPNP weights of (s, t), (s, d), and (d, t) are summarized in Table 1.

As shown in Table 1, it is obvious that, according to Eqs. (1) to (3), the various weights of (s, t) are W_{SPC} $(s, t) = n_1 \cdot m_1$, W_{SPLC} $(s, t) = n_2 \cdot m_1$, and W_{SPNP} $(s, t) = n_2 \cdot m_2$. It is also straightforward to see that, under SPC, W_{SPC} $(s, t) = W_{SPC}$ $(s, d) = W_{SPC}$ $(d, t) = n_1 \cdot m_1$ as (s, t), (s, d), and (d, t) involve the same sets of sources and sinks, as shown in the second column of Table 1. Similarly, under SPLC, (s, t) and (s, d) share identical sets of preceding nodes and sinks, and, therefore, W_{SPLC} $(s, t) = W_{SPLC}$ $(s, d) = n_2 \cdot m_1$.

As to (d, t) under SPLC and SPNP, W_{SPLC} $(d, t) = (n_2 + 1) \cdot m_1$ and W_{SPNP} $(d, t) = (n_2 + 1) \cdot m_2$, where the $(n_2 + 1)$ factor is resulted from that, in addition to the n_2 paths from *preceding* $(s, t) =$ *preceding* (s, d) reaching d then t, an additional path (d, t) has to be counted. Similarly, for (s, d) under SPNP, W_{SPNP} $(s, d) = n_2 \cdot (m_2 + 1)$, where the $(m_2 + 1)$ factor is also due to that the additional path (s, d) has to be added to the m_2 paths out of s through d to *succeeding* $(s, t) =$ *succeeding* (d, t).

Table 1 The SPC, SPLC, and SPNP weights for the arcs of Fig. 1

Arc	W_{SPC}	W_{SPLC}	W_{SPNP}
(s, t)	$n_1 \cdot m_1$	$n_2 \cdot m_1$	$n_2 \cdot m_2$
(s, d)	$n_1 \cdot m_1$	$n_2 \cdot m_1$	$n_2 \cdot (m_2 + 1)$
(d, t)	$n_1 \cdot m_1$	$(n_2 + 1) \cdot m_1$	$(n_2 + 1) \cdot m_2$

To see how the various path search algorithms perform with the arc weights of Table 1, it is assumed that at least one of the MPs selected by these path search algorithms would extend from sub-net 1 to sub-net 2, and the MP then must run through the nodes *s* and *t*.

If global standard search is applied, MPA will always pick the longer path through (*s*, *d*) and (*d*, *t*) as they together contribute a higher combined weight than that of (*s*, *t*), regardless of which weight assignment algorithm is adopted. In other words, MPA using global standard search algorithm indeed favors the longer paths regardless of the weight assignment used.

MPA behaves less greedy under local search, compared to global search. With the local forward search algorithm, the MP is developed stage by stage from sub-net 1 towards the sub-net 2. As the forward search reaches *s*, if the arc weights are assigned using SPC or SPLC algorithm, both (*s*, *t*) and (*s*, *d*) will be traced as they are of the same weight. As such, both the longer and shorter paths are part of the resulted MP, regardless of the set tolerance. However, if (*s*, *t*) and (*s*, *d*) are weighed by SPNP and the tolerance is less than $(1/(m_2+1)=1-m_2/(m_2+1))$, MPA will choose the higher-weighted (*s*, *d*), and the MP runs through the longer path. However, if the tolerance is set at least $(1/(m_2+1))$, both paths would be retained.

With the local backward search algorithm, the MP is developed stage by stage in reverse from sub-net 2 to the sub-net 1. As the search backtracks to *t*, if (*s*, *t*) and (*d*, *t*) are assigned with SPC weights, (*s*, *t*) and (*d*, *t*) are both traced, and the longer and shorter paths are retained in the final MP. If (*s*, *t*) and (*d*, *t*) are assigned with SPLC or SPNP weights, however, MPA will choose the higher-weighted (*d*, *t*) and go for the longer path if the tolerance is less than $(1/(n_2+1))$.

The MPA preference is more complicated under global and local key-route search algorithms as there may be multiple key routes dispersed in sub-net 1, in sub-net 2, or between sub-net 1 and sub-net 2. For simplicity's sake, the following discussion assumes that (1) global or local key-route 1 is applied (i.e., the key routes are restricted to the highest weighted arcs), and (2) these key routes are located altogether only in sub-net 1, in sub-net 2, or between them.

If the highest weighted arc or arcs are among (*s*, *t*), (*s*, *d*), and (*d*, *t*), for SPC arc weights, all three arcs would be chosen as key routes as they are identically weighted. The rest of the MP then would be developed therefrom into sub-net 1 and sub-net 2 following either the local search or global search manner. As such, both longer and shorter paths between *s* and *t* are retained as part of the MP. However, for arc weights by SPLC, only (*d*, *t*) is chosen as the key route whereas, if weights are assigned by SPNP, (*s*, *d*), (*d*, *t*), or both may be key routes, depending on the values n_2 and m_2. Therefore, whether weights are by SPLC or SPNP or whether a non-zero tolerance is set, it is the longer path between *s* and *t* that is included in the MP.

If the highest weighted arc or arcs are within either sub-net 1 or sub-net 2, and if global key-route search is employed, these key routes are developed into one or more MPs. Assuming that one of the MPs would cross *s* and *t*, the MPA's choice of paths between *s* and *t* would be just like that under the global standard search algorithm, where the longer path between *s* and *t* is always chosen, regardless of the weight assignment algorithm used.

If the highest weighted arc or arcs are within sub-net 1, the one or more MPs developed by the local key-route search will extend from these key routes. Assuming that one of the

MPs under development reaches *s*, MPA would perform just like the local forward search. Similarly, if the highest weighted arc or arcs are within sub-net 2 and if one of the MPs under development reaches *t*, MPA would be behave just like the local backward search.

Summary

Regarding the question whether MPA always picks the paths consisting of the higher numbers of arcs among the multiple paths between a pair of nodes, Table 2 summarizes the MPA's choices under various combinations of the weight assignment and path search algorithms explored in the previous section.

As can be seen from Table 2, the claim that MPA favors the longer paths over the shorter ones is not always valid. Even though MPA never goes only for the shorter paths, it may retain them, along with the longer ones, as part of the MPs under some combinations of the algorithms.

Table 2 reveals that (1) global standard search algorithm always picks the longer paths regardless of whether the arc weights are set by SPC, SPLC, or SPNP; (2) global key-route also always follows the longer paths and the only exception may occur when it is used with SPC weights; (3) SPNP arc weights with local search algorithms generally lead to the longer paths' being chosen, except when a tolerance large enough is set; (4) SPC arc weights with local search algorithms always retain both the longer and shorter paths; and (5) the combination of SPLC and local search would reveal the most diverse behavior, depending on the variant used and the search parameter set.

Whether a longer MP or a shorter one is "better" is not the focus of the present study. However, in the author's opinion, this debatable question can be addressed from two perspectives. Firstly, if two entirely or largely different MPs are derived by different MPA approaches from a same citation network, these MPs should be judged by how representative they are in reflecting a field's evolution, rather than simply by their path lengths.

On the other hand, if these MPs share some common nodes and reveal generally consistent trends, the MPs can actually complement each other in providing a more comprehensive evolution trajectory. Figure 3 depicts two fictitious MPs, in which the shorter MP consists of white and black nodes, the longer one involves grey and black nodes, and the black nodes are common to both MPs.

Then, for example, the grey nodes of the longer MP seem to offer some additional and possible development alternatives while not deviating from the main trend set by the black nodes, like the grey nodes 60 to 65, or the grey nodes 21 to 24. In addition, the grey nodes from the longer MP between nodes 42 and 52, and between nodes 52 and 60 seem to provide finer details about the development from node 42 to node 60. Again, one may argue that it is redundant and distracting to have these grey nodes when the shorter one already reveals the main skeleton of evolution trend.

Despite that a consensus is yet to be reached as outlined above, the author believes that a longer MP does not necessarily imply its inferiority to a shorter one, and the longer MP may offer some additional insight that may be missing from the shorter MP. The author also believes that the key issue regarding whether a longer MP or a shorter one is "better" should lie in their representativeness to the field's development, instead of their lengths. However, how to determine the representativeness of a MP will be left for future studies.

Table 2 Summary of MPA preference for the simplified model of Fig. 1

Search algorithm			SPC weight	SPLC weight	SPNP weight
Global	Standard		Longer path	Longer path	Longer path
	Key-route 1	Key routes between s and t	Both longer and shorter paths	Longer path	Longer path
		Key routes in sub-net 1	Longer path	Longer path	Longer path
		Key routes in sub-net 2	Longer path	Longer path	Longer path
Local	Forward		Both longer and shorter paths	Both longer and shorter paths	Longer path when tolerance $< 1/(m_2+1)$; otherwise, both longer and shorter paths
	Backward		Both longer and shorter paths	Longer path when tolerance $< 1\ /(n_2+1)$; otherwise, both longer and shorter paths	Longer path when tolerance $< 1/(n_2+1)$; otherwise, both longer and shorter paths
	Key-route 1	Key routes between s and t	Both longer and shorter paths	Longer path	Longer path
		Key routes in sub-net 1	Both longer and shorter paths	Both longer and shorter paths	Longer path when tolerance $< 1/(m_2+1)$; otherwise, both longer and shorter paths
		Key routes in sub-net 2	Both longer and shorter paths	Longer path when tolerance $< 1/(n_2+1)$; otherwise, both longer and shorter paths	Longer path when tolerance $< 1/(n_2+1)$; otherwise, both longer and shorter paths

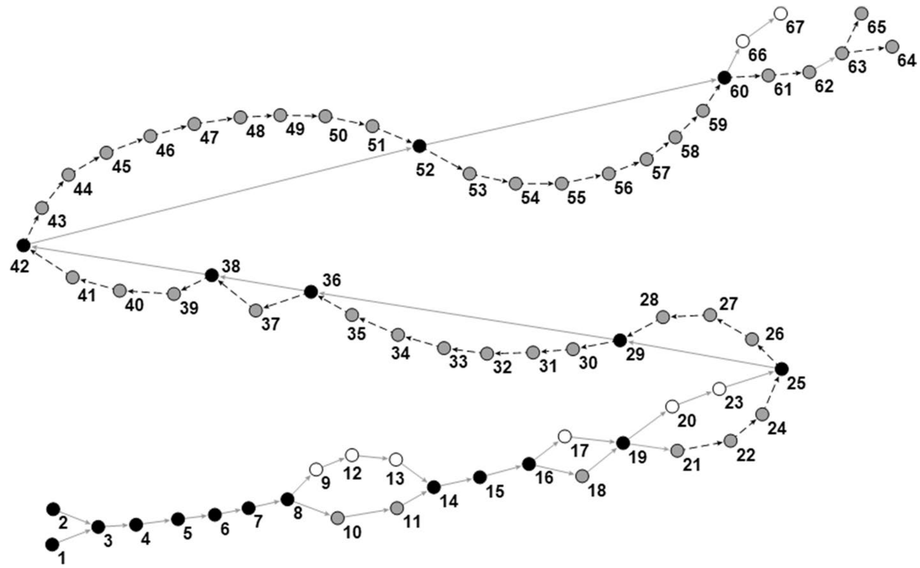

Fig. 3 Two fictitious MPs, one involving white and black nodes and the other one grey and black nodes

Acknowledgements The author expresses his appreciation to the anonymous reviewers for their careful examination of the manuscript and their helpful comments and suggestions. This work was financially supported by the Center for Research in Econometric Theory and Applications (Grant No. 111L900204) from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project, the Universities and Colleges Humanities and Social Sciences Benchmarking Project (Grant No. 111L9A002) by the Ministry of Education (MOE) in Taiwan, and by the Ministry of Science and Technology (MOST), Taiwan, under Grant Nos. 111-2634-F-002-018- and 110-2221-E-011-141-.

Declarations

Conflict of interest The author declares that he has no conflict of interest.

References

Batagelj, V. (2003). Efficient algorithms for citation network analysis. Preprint at http://arXiv.org/cs/0309023

Batagelj, V., & Mrvar, A. (1998). Pajek—Program for large network analysis. *Connections, 21*(2), 47–57.

Chen, L., Xu, S., Zhu, L., Zhang, J., Xu, H., & Yang, G. (2022). A semantic main path analysis method to identify multiple developmental trajectories. *Journal of Informetrics, 16*(2), 101281.

Choi, C., & Park, Y. (2009). Monitoring the organic structure of technology based on the patent development paths. *Technological Forecasting and Social Change, 76*(6), 754–768.

De Nooy, W., Mrvar, A., & Batagelj, V. (2018). *Exploratory social network analysis with Pajek: Revised and expanded edition for updated software* (Vol. 46). Cambridge University Press.

Hummon, N. P., & Doreian, P. (1989). Connectivity in a citation network: The development of DNA theory. *Social Networks, 11*(1), 39–63.

Jiang, X., Zhu, X., & Chen, J. (2020). Main path analysis on cyclic citation networks. *Journal of the Association for Information Science and Technology, 71*(5), 578–595.

Kim, E. H., Jeong, Y. K., Kim, Y., & Song, M. (2022). Exploring scientific trajectories of a large-scale dataset using topic-integrated path extraction. *Journal of Informetrics, 16*(1), 101242.

Kuan, C. H., Chen, D. Z., & Huang, M. H. (2019). Bibliographically coupled patents: Their temporal pattern and combined relevance. *Journal of Informetrics, 13*(4), 100978.

Liu, J. S., & Lu, L. Y. (2012). An integrated approach for main path analysis: Development of the Hirsch index as an example. *Journal of the American Society for Information Science and Technology, 63*(3), 528–542.

Lu, Z., Ma, Y., & Song, L. (2021). Patent citation network analysis based on improved main path analysis: Mapping key technology trajectory. In *International conference on artificial intelligence and security* (pp. 158–171). Springer.

Park, H., & Magee, C. L. (2017). Tracing technological development trajectories: A genetic knowledge persistence-based main path approach. *PLoS ONE, 12*(1), e0170895.

Šubelj, L., Waltman, L., Traag, V., & van Eck, N. J. (2020). Intermediacy of publications. *Royal Society Open Science, 7*(1), 190207.

Xu, S., Wang, C., An, X., Hao, L., & Yang, G. (2022). A novel developmental trajectory discovery approach by integrating main path analysis and intermediacy. *Journal of Information Science*. https://doi.org/10.1177/01655515221101835

Yoon, S., Mun, C., Raghavan, N., Hwang, D., Kim, S., & Park, H. (2020). Hierarchical main path analysis to identify decompositional multi-knowledge trajectories. *Journal of Knowledge Management, 25*(2), 454–476.

Yu, D., & Pan, T. (2021). Tracing the main path of interdisciplinary research considering citation preference: A case from blockchain domain. *Journal of Informetrics, 15*(2), 101136.