COMPLETE MULTIPARTITE GRAPHS ARE PANSOPHICAL

JEFFE BOATS, LAZAROS KIKAS

ABSTRACT. Given a graph G, we are interested in finding disjoint paths for a given set of distinct pairs of vertices. In 2017, we formally defined a new parameter, the pansophy of G, in the context of the disjoint path problem. In this paper, we construct a method to determine the pansophy of any complete bipartite graph, and then generalize the method to compute the pansophy any complete multipartite graph. We close with future research directions.

Keywords: Pansophy, interconnection networks, graphs, algorithms, vertex disjoint paths

1. MOTIVATION

In the study of interconnection networks, we are interested the simultaneous routing of disjoint paths in a graph, as this avoids queuing and leads to faster processing. Much research has been done, particularly for Cayley graphs, on the $k-$Disjoint Path Problem: given k pairs of vertices (s_1, t_1), (s_2, t_2),. . .,(s_k, t_k), do there exist disjoint paths in the graph for each pair of vertices? If we can find disjoint paths for any collection of k pairs of vertices, then G satisfies the **k-Disjoint Path Property**. See [1], [2], [4], [5], [6].

For the benefit of those writing algorithms for routing communications within networks, we would like a way of measuring how well a routing algorithm performs. The k-disjoint path property guarantees k paths for any choice of vertex pairs, but more than k will often be possible. For a given assignment of vertex pairs, the **maximal routing volume** is defined as the largest number of disjoint paths possible.

In the beginning, 2017, Boats and Kikas defined **pansophy** as the expected value for maximal routing value in a graph, which means that the pansophy of a graph is the best possible performance or any any routing algorithm. [3] This makes it an ideal measuring stick for evaluating algorithms, but this requires us to calculate the pansophy of a graph, or at least well approximate it. For this reason, we are motivated to explore the pansophies of entire classes of graphs. In this paper, we will construct a means of calculating the pansophy of any complete multipartite graph.

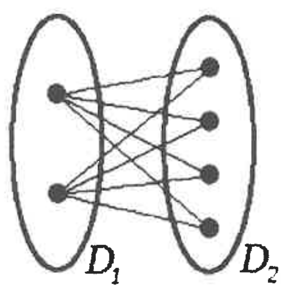
2. Definitions

Let G be a graph, and let $\Omega = \lfloor \frac{|V(G)|}{2} \rfloor$, the maximum possible number of paths which could be simultaneously routed in G. We define the **pansophy** of G, $\Psi(G)$, as the expected number of simultaneously routed disjoint paths, given foreknowledge of upcoming pairs of vertices; we derive a formula for computing $\Psi(G)$ as follows.

In [3] we derive the following. Let p_i be the probability that i randomly assigned terminal pairs $(s_1, t_1), \ldots, (s_i, t_i)$ can be disjointly routed. The probability $\phi(i)$ that a random assignment of $(s_1, t_1), \ldots, (s_\Omega, t_\Omega)$ can have i paths routed disjointly, but not $i+1$, is $\phi(i) = p_i - p_{i+1}$. Then

$$\Psi(G) = \sum_{i=1}^{\Omega} i\ \phi(i) = \sum_{i=1}^{\Omega} i\ (p_1 - p_{i+1}) = \ldots = \sum_{i=1}^{\Omega} p_i .$$

A demonstration: in calculating the pansophy of $K_{2,4}$, we quickly see that $p_1 = 1$ (because the graph is connected) and $p_3 = 0$ (because any path must use at least one vertex from each partition), so p_2 is where the effort lies.



For p_2, notice that the only two cases where we can't connect two paths is if the two vertices in D_1 are the pair (s_1, t_1) or the pair (s_2, t_2). In the former case, the first path leaves a remaining graph that is three disconnected vertices; the latter case leaves a remaining graph isomorphic to P_3, but the first path must use one of the (s_2, t_2) pair, eliminating the possibility of a second path.

Each of these cases happens with a $\frac{1}{15}$ probability.

Thus $p_2 = 1 - 2(\frac{1}{15}) = \frac{13}{15}$, and so $\Psi(K_{2,4}) = 1 + \frac{13}{15} + 0 = \frac{28}{15}$. □

Computing the pansophy of a graph, in general, is very difficult. Some clases of graphs, however, lend themselves to more eficient analysis. A class of graphs Γ is **pansophical** if, for any $G \in \Gamma$, the pansophy $\Psi(G)$ is computable in polynomial time:

(1) by explicit formula involving the graphs' parameters as variables;
(2) by iterative algorithm, based on some of the graph's parameters.

In this paper, we will demonstrate that the class of multipartite graphs is pansophical by iterative algorithm. We begin my proving bipartite graphs are pansophical, and then generalize the result for multipartite graphs.

3. Original Approach for Bipartite Graphs

Let a complete bipartite graph K_{m_1,m_2} be given, and let its two partitions be labeled D_1 and D_2, with $m_1 = |D_1|$ and $m_2 = |D_2|$ respectively. Define $M = \sum_i m_i = m_1 + m_2$. We begin the algorithmic approach to finding its pansophy with two base cases.

First, if either partition has zero vertices, then the graph has no edges and the pansophy is trivially zero. Second, if the partitions are non-empty but one of them has only one vertex, then the graph is a star graph, and the pansophy must be one, because the first path must use the center vertex and disconect every remaining vertex in the graph. Therefore,

$$\Psi(K_{0,m_2}) = \Psi(K_{m_1,0}) = 0 \qquad \textit{and} \qquad \Psi(K_{1,m_2}) = \Psi(K_{m_1,1}) = 1 .$$

Suppose now that $m_1 \geq 2$ and $m_2 \geq 2$, and let vertices s_1 and t_1 in K_{m_1,m_2} be given. Consider the shortest path which can be routed between them. If s_1 and t_1 are in different partitions, then they are adjacent, and the shortest path is the edge connecting them. On the other hand, if they are in the same partition, then the shortest path involves a "jaunt."

Definition Given s_i and t_i both in the same subgraph, a jaunt from s_i to t_i is a two-edge path whose intermediate vertex is external to the subgraph.

For s_1 and t_1 in opposite partitions, we remove these vertices and all edges adjacent them, leaving a remainder graph isomorphic to K_{m_1-1,m_2-1}. This happens with probability:

$$\lambda_{1,2} = P(s_1, t_1 \textit{ adjacent}) = \frac{2C_{m_1,1}C_{m_2,1}}{C_{M,2}} = \ldots = \frac{2m_1m_2}{M(M-1)} .$$

If s_1 and t_1 are in the same partition, say D_1, we remove these two vertices plus one vertex in D_2, and all edges adjacent to them, leaving a remainder graph isomorphic to K_{m_1-2,m_2-1}. This happens with probability:

$$\lambda_{1,1} = P(s_1, t_1 \in D_1) = \frac{C_{m_1,2}}{C_{M,2}} = \ldots = \frac{m_1(m_1+1)}{M(M-1)} .$$

Lastly, if s_1 and t_1 are in the second partition D_2, then we remove these vertices plus one vertex in D_2, and all edges adjacent them, leaving a remainder graph isomorphic to K_{m_1-1,m_2-2}. This happens with probability:

$$\lambda_{2,2} = P(s_1, t_1 \in D_2) = \frac{C_{m_2,2}}{C_{M,2}} = \ldots = \frac{m_2(m_2+1)}{M(M-1)} .$$

Using these probabilities, we can construct a recursion formula which derives $\Psi(K_{m_1,m_2})$ from smaller graphs of the same class. Its pansophy is 1, for the initial path we can construct, plus a weighted average of the pansophies of the possible remainder graphs:

$$\Psi(K_{m_1,m_2}) = 1 + \frac{2m_1m_2}{M(M-1)}\ \Psi(K_{m_1-1,m_2-1})$$
$$+ \frac{m_1(m_1-1)}{M(M-1)}\ \Psi(K_{m_1-2,m_2-1}) + \frac{m_2(m_2-1)}{M(M-1)}\ \Psi(K_{m_1-1,m_2-2}) .$$

And it would seem we have a recursive formula for the pansophies of any bipartite graph, building steadily up from our base cases. But the algorithm above returns $\Psi(K_{2,4}) = \frac{29}{15}$, when we saw earlier the correct result is $\frac{28}{15}$. It turns out there is one slight flaw that has been overlooked, and which we must address.

What went wrong? The algorithm missed the drop of $\frac{1}{15}$ for the case where D_1 consists of exactly s_2 and t_2, so that creating the first path leaves a connected remainder graph, but must burn the second path in routing the first one! To correct the formula, we must consider a new parameter.

4. Jaunting to Free Vertices

Definition For a given (s_i, t_i) assignment within a graph G, a free vertex is one whose path can no longer be connected, and hence it is freely available for use in connecting other paths with no risk of lowering the maximal routing volume.

For example, in any graph with an odd number of vertices, given any assignment of pairs $(s_1,t_1),\ldots,(s_\Omega,t_\Omega)$, there will be an odd vertex, say $s_{\Omega+1}$, which has no mate. It can be part of no path, and hence is a free vertex from the beginning.

Later on, after some paths have been routed, those paths may have used vertices which are starts or finishes for paths of a higher index. The mates of those used vertices become free vertices, for their assigned paths have been "burned" in the process of connecting earlier paths.

Definition The state of the bipartite graph K_{m_1,m_2} is denoted $K^z_{m_1,m_2}$, where z refers to the number of free vertices available.

For example, the initial state of $K_{2,4}$ is in fact $K^2_{2,4}$, since the third path can never be routed, making s_3 and t_3 free vertices from the start.

What if we need an intermediate vertex to route s^* to t^*, but there are no free vertices? Then we must burn a possible path of higher index. To maximize routing volume, we choose the vertex with the highest index – this naturally creates a "jaunting list" of vertices, which initially looks like:

$$s_{\Omega+1} \succ t_\Omega \succeq s_\Omega \succ t_{\Omega-1} \succeq s_{\Omega-1} \succ \ldots \succ t_2 \succeq s_2 \succ t_1 \succeq s_1$$

We remove vertices from the list as they are used, and always select the available vertex of highest (leftmost) preference.

Let s^* and t^* be in the same partition D_i, with $|D_i| = m_i \geq 2$. To route this path, we must jaunt to the opposite partition. The effect this has on the pansophy of the remainder graph depends on whether we need to burn a later path in the process, and that depends on whether any free vertices are in the opposite partition. We define the following probability which will help us determine whether a free vertex is available for us to jaunt through, or whether we must burn one or more later paths to create one.

The probability that the first J vertices in the current jaunting list are also in D_i is:

$$\delta_{i,J} = \begin{cases} 0 & ;\ if\ m_i = 2\ \ or\ \ J > m_i - 2\ ; \\ 1 & ;\ if\ J = 0\ ; \\ \frac{C_{m_i-2,J}}{C_{M-2,J}} & ;\ if\ 0 < J \leq m_i - 2\ . \end{cases}$$

5. Algorithm for Pansophy of Bipartite Graphs

Beginning with the example of $K^2_{2,4}$, the probability $\lambda_{2,2}$ that both s_1 and t_1 are in D_2 is associated with a remainder graph $K_{1,2}$ after that path is routed by jaunting to a vertex in D_1, but that is a not sufficient analysis.

We must consider the two possible states of that remainder graph, $K^1_{1,2}$ and $K^3_{1,2}$. The former is the case where a free vertex (or both) is in D_1, so that the number of free vertices decreases by one and no later paths must be burned. The latter state is the case where neither free vertex was available for a jaunt, as both were in D_2, and so a later path (the second) must be burned, creating two new free vertices but then using one of them, which is why the number of free variables increases to three.

$$\begin{aligned}\Psi(K^2_{2,4}) &= 1 + \lambda_{1,2}\ \Psi(K^2_{1,3}) \\ &\quad + \lambda_{1,1}\ \Psi(K^1_{0,3}) \\ &\quad + \lambda_{2,2}\ [\ (1-\delta_{2,2})\ \Psi(K^1_{1,2}) + \delta_{2,2}\ \Psi(K^3_{1,2})\ ] \\ &= 1 + \tfrac{8}{15}(1) \\ &\quad + \tfrac{1}{15}(0) \\ &\quad + \tfrac{6}{15}[(1-\tfrac{1}{6})(1) + (\tfrac{1}{6})(0)] \\ &= 1 + \tfrac{8}{15} + 0 + \tfrac{5}{15} = \tfrac{28}{15}\end{aligned}$$

And the correct answer emerges. Note that the $\delta_{2,2}$ probabiliy that a later path needed burning was as far as we had to take it, since the small size of the graph guaranteed that one burnt path would be enough to produce a free vertex.

When we deal with larger graphs, we have to take in to account the possibility that more than one path might need to be burnt. The good news is that a given partition D_i is limited in size, and there are only so many places for vertices on the jaunting list to hide. A free vertex outside of D_i must be found after burning at most $\sigma_i = \lceil \frac{m_i - J - 1}{2} \rceil$ paths. Summing over the possibilities, we arrive at the corrected algorithm formula:

$$\begin{aligned}\Psi(K^J_{m_1,m_2}) \;=\; 1 \;&+\; \lambda_{1,2}\ \Psi(K^J_{m_1-1,m_2-1}) \\ &+\; \lambda_{1,1}\ [\ (1-\delta_{1,J})\ \Psi(K^{J-1}_{m_1-2,m_2-1}) \\ &\qquad +\ \textstyle\sum_{j=1}^{\sigma_1}\ (\delta_{1,J+2j-2} - \delta_{1,J+2j})\ \Psi(K^{J+2j-1}_{m_1-2,m_2-1})\ ] \\ &+\; \lambda_{2,2}\ [\ (1-\delta_{2,J})\ \Psi(K^{J-1}_{m_1-1,m_2-2}) \\ &\qquad +\ \textstyle\sum_{j=1}^{\sigma_2}\ (\delta_{2,J+2j-2} - \delta_{2,J+2j})\ \Psi(K^{J+2j-1}_{m_1-1,m_2-2})\ ]\end{aligned}$$

The base cases need alteration as well, as they must take into account the parameter of free vertices. We observe that no paths can be routed if all remaining vertices are free, nor if all but one are free. To the previous base cases, we append:

$$\Psi(K^M_{m_1,m_2}) = \Psi(K^{M-1}_{m_1,m_2}) = 0\ .$$

6. Generalization to Multipartite Graphs

Nearly everything in the derivation of the complete bipartite graph algorithm applies to complete multipartite graphs. The only difference is that, when completing a path requires a jaunt, there may be choice of which partition to jaunt to, depending on how many free vertices there are, and where they are. The computation of the probablities for the various contingencies amount to standard combinatorial calculations, but the organization of these contingencies will require considerable care.

Let G be a multipartite graph, and let $G^J_{a_1,\ldots,a_n}$ denote the state of the remainder graph of G which has J free vertices and where a vertex has been removed from partition D_{a_i} for each $i = 1,\ldots,n$. With this notation, we can rewrite the bipartite formula into a multipartite formula, except that the asterisk seen in the last term refers to a partition index that is undetermined:

$$\begin{aligned}\Psi(G^J) \;=\; 1 \;&+\; \textstyle\sum_i \sum_{k\neq i} \lambda_{i,k}\ \Psi(G^J_{i,k}) \\ &+\; \textstyle\sum_i \lambda_{i,i}\ \{\ (1-\delta_{i,J})\ \Psi(G^{J-1}_{i,i,*}) \\ &\qquad +\ \textstyle\sum_{j=1}^{\sigma_i}\ (\delta_{i,J+2j-2} - \delta_{i,J+2j})\ \Psi(G^{J+2j-1}_{i,i,*})\ \}\ .\end{aligned}$$

The asterisk refers to the fact that there will sometimes be a choice from which partition the free vertex is chosen; the pansophy $\Psi(G^{J+2j-1}_{i,i,*})$ is a weighted average of all possible remainder pansophies $\Psi(G^{J+2j-1}_{i,i,k})$, $\forall\, k \neq i$.

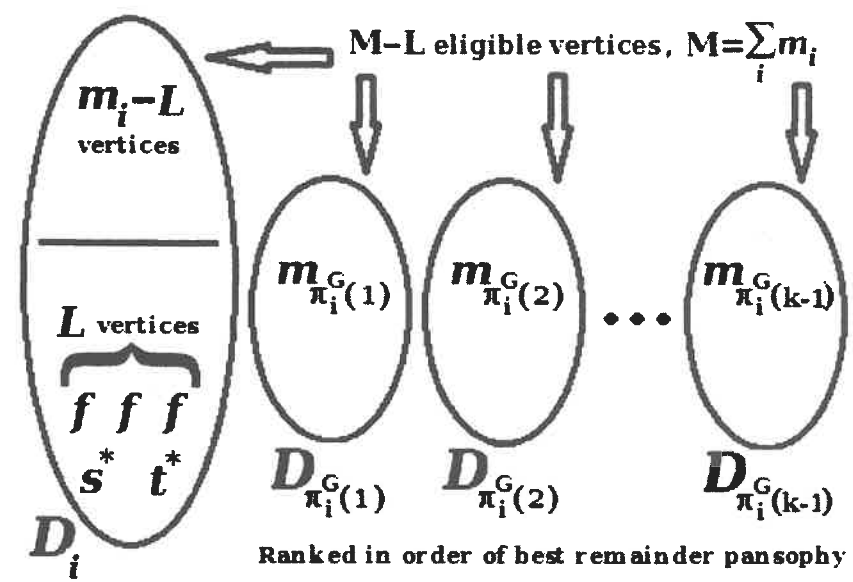
We will list these remainder pansophies in descending order. Define π^G_i to be the permutation of $K = \{1, 2, \ldots, k\}$ that maps i to k, and maps all other $i \in K$ to the order in which its corresponding remainder graph appears. In other words,

$$\pi^G_i(a) = \begin{cases} k & ;\ \ a = i \\ b & ;\ \ \Psi(G_{i,i,b}) \textit{ is the } a^{\underline{th}} \textit{ highest pansophy} \\ & \qquad \textit{of all possible remainder graphs}\end{cases}$$

The $k-1$ possible remainder graphs would then be listed, in descending order, as follows:

$$\Psi(G_{i,i,\pi^G_i(1)}) \geq \Psi(G_{i,i,\pi^G_i(2)}) \geq \cdots \geq \Psi(G_{i,i,\pi^G_i(k-1)})$$

When considering which partition to use, we first try all free vertices. If all of them are in D_i, and thus ineligible, we burn a late (s_i, t_i) pair and try again with these two new vertices. We burn as many paths as are necessary until we find a free vertex that works. If at any point we have more than one that works, we take the one for which the remainder graph will have the best pansophy.



Define $g^G_i(L, h, \pi^G_i(a))$ to be the probability that, in G, given L vertices in D_i which are ineligible for consideration (because they are s^*, t^*, or free vertices that are unusable), and h free vertices that are eligible, that there is an eligible free vertex in $D_{\pi_i^G(a)}$, but not in any partition which yields a better remainder graph pansophy.

For example, if we want to route from s^* to t^* with both in D_i, and we have J free vertices randomly distributed through the rest of G, then $L = 2$ because only s^* and t^* are used or ineligible, and $h = J$ because all of the J free vertices are eligible for consideration. We would denote with $g_i^G(2, J, \pi_i^G(a))$ the probability that the best available choice of partition to jaunt to is the one with the ath best remainder graph.

Now suppose none of the J free vertices worked, because they were all also in D_i (depicted by f's in the above diagram). In this case, $L = 2 + J$, and after we burn a late (s_i, t_i) pair to create two new eligible vertices, we say $h = 2$. At this point, the probability that the best available choice of partition is the ath is denoted $g_i^G(2 + J, 2, \pi_i^G(a))$.

The values of these probabilities are computed combinatorially; they are simply the probabilities of h eligible vertices being in decreasing sizes of graphs as we first eliminate the best partition choice, then the second best, et cetera. If we define $s(t) = \sum_{a=1}^{t} m_{\pi_i^G(a)}$ to be the number of vertices in the best t partitions, then the formulas are:

$$g_i^G(L, h, \pi_i^G(1)) = 1 - \frac{C_{M-s(1)-L,h}}{C_{M-L,h}} ;$$

$$\text{For } 2 \le b \le k-1 , \quad g_i^G(L, h, \pi_i^G(b)) = \frac{C_{M-s(b)-L,h} - C_{M-s(b-1)-L,h}}{C_{M-L,h}} .$$

In the multipartite pansophy algorithm, these probabilities generalize the $\delta_{i,J}$ terms. Since $1 - \delta_{i,J}$ is the probability that a free vertex is available for immediate jaunting without the need to burn a late pair, we have:

$$\sum_{a=1}^{k-1} g_i^G(2, J, \pi_i^G(a)) = 1 - \delta_{i,J} .$$

Simillarly, since $\delta_{i,J}$ is the probability that we are required to burn at least one pair, but we must eventually find a good jaunting partition by burning enough pairs (up to a maximum of σ_i), we have:

$$\sum_{j=1}^{\sigma_i} \left[\sum_{a=1}^{k-1} g_i^G(J + 2j, 2, \pi_i^G(a)) \right] = \delta_{i,J} .$$

Infusing these summations with the appropriate remainder graph pansophies, we arrive at the general formula or the pansophy of a multipartite graph G, calculated from the known pansophies of smaller multipartite graphs:

$$\begin{aligned}\Psi(G^J) \;=\; 1 \;&+\; \textstyle\sum_i \sum_{k \ne i} \lambda_{i,k}\, \Psi(G^J_{i,k}) \\ &+\; \textstyle\sum_i \lambda_{i,i} \, \{ \sum_{a=1}^{k-1} g_i^G(2, J, \pi_i^G(a)) \, \Psi(G^{J-1}_{i,i,\pi_i^G(a)}) \\ &\quad+\; \textstyle\sum_{j=1}^{\sigma_i} \sum_{a=1}^{k-1} g_i^G(J+2j, 2, \pi_i^G(a)) \, \Psi(G^{J+2j-1}_{i,i,\pi_i^G(a)}) \} .\end{aligned}$$

7. Future Directions

We will soon be engaging in undergraduate research by having a talented student encode the above algorithm, likely in C++ or Python, to compute the pansophies for a large library of complete multipartite graphs. As the formula is very complicated, it will be in our interest to try to find a more reasonable formula, if possible – something much simpler which closely approximates the results above.

It is likely (though not well explored yet) that the pansophy of a dense graph is "close" to that of a graph which differs by only a few edges. For the routing algorithms at use in those communication networks which are "close" to being multiparatite, having a reasonable approximation formula for the pansophy of multipartite graphs would be beneficial in measuring algorithm efficiency.

Finally, we will continue to explore different classes of graphs, and particularly we will focus on Cayley graphs which have been recommended by past Disjoint Path Problem research. The goal is to prove them pansophical, or at least to find tight bounds on their pansophies.

References

[1] E.Cheng, L.D. Kikas, and S.Kruk. A disjoint path problem in the alternating group graph. *Congressus Numerantium*, 175:117-159, 2005.

[2] E.Cheng and M. Lipman. Disjoint paths in split-stars. *Congressus Numerantium*, 137:47-63, 1999.

[3] J.Boats and L.D.Kikas. The pansophy of a graph. *Congressus Numerantium*, 229:125-134, 2017.

[4] J.Boats, L.D.Kikas, and J.Oleksik. Algorithm for finding disjoint paths in the alternating group graph. *Congressus Numerantium*, 181:97-109, 2006.

[5] J.Boats, L.D.Kikas, and J.Oleksik. The nova graph: More disjoint paths with minimal graph augmentation. *Congressus Numerantium*, 184:71-83, 2007.

[6] J.Boats, L.D.Kikas, and J.Oleksik. An algebraic approach for finding disjoint paths in the alternating group graph. *The Journal of Combinatorial Mathematics and Combinatorial Computing*, 64:109-119, 2008.

Jeffe Boats, Corresponding Author, Department of Mathematics, University of Detroit Mercy, Detroit, MI, 48221-3038, USA

E-mail address: boatsjj@udmercy.edu

Lazaros Kikas, Department of Mathematics , University of Detroit Mercy, Detroit, MI, 48221-3038, USA
E-mail address: kikasld@udmercy.edu