The Dilemma with Romance as a Business and How Tinder Solves It

Author: Clayton H. Pruitt[1]
Mentor: Dr. Mary Lynn Reed [2]
School of Mathematical Sciences, Rochester Institute of Technology

Abstract

It is paradoxical for an optimal online dating service to exist since a service which pairs all of its users will have no customers. As such, Tinder's algorithms must balance the user's desire to be paired in a relationship with Tinder's need for enough users to run a successful business. However, these algorithms are proprietary and confidential, requiring an outside analysis in order to determine how they function. We start by observing applications of the stable marriage problem in set theory, graph theory, and network science to determine how an online dating service can optimally match users. We then introduce the concept of reciprocal recommender systems to strengthen the connection between the stable marriage problem and Tinder. These insights are used to create a model for the optimal online dating service to show that Tinder is capable of performing optimally, but must de-optimize itself to prevent pairing all users in relationships. It is determined that premium features are strong de-optimization methods as they keep users within Tinder's user base while also increasing the service's revenue.

1 Introduction

An *online dating service* is a platform designed to help a person, or *user*, find their ideal partner. Users will provide the service with personal information, such as age, gender, or sexual preference [3], that will constitute the user's *profile*. Profiles are then input into a *recommender algorithm* [1] to determine which users should be recommended to one another based off the value, or *utility*, of the recommendation. If two users who have been recommended to each other express reciprocal interest, a channel of communication is opened to allow further exploration of interest [3]. These services have proven to be quite successful, spurring a rapid market growth. However, analysts predict that growth will start to decrease significantly in the near future [6].

One possible explanation is that users who form relationships no longer have use of an online dating service. With this in mind, it could be stated that online dating services are self-destructive. By fulfilling their advertised purpose of forming relationships, they will diminish the size of their target audience with every success. Without enough users, the service will be unable to operate as a successful business. Thus, there appears to be a paradox that prevents an *optimal online dating service* from existing. Online dating services must find ways to *de-optimize* if they wish to continue

[1]chp4145@rit.edu
[2]mlrsma@rit.edu

as a business. However, users will also cease to use a service that performs poorly. There must then be a balance between service profitability and user satisfaction. This balance appears to be highly influential in algorithms utilized by the online dating service Tinder.

Tinder has been described as a mobile dating app that enables the discovery of nearby singles [9]. As of 2020, the service is utilized by fifty-seven million users globally across one-hundred and ninety countries [5]. An initial set of features are provided free of charge, but additional *premium features* are offered at a cost [12].

A simplistic summary of Tinder's business model can be made with this information. Introduce users to a limited selection of the service's features and then tempt them to pay for additional features that claim to improve the user's chances of finding success [3]. This model appears to be quite successful, as the service generated over a billion dollars in revenue in 2019. In fact, we will find that this model is Tinder's strongest de-optimizer as premium features encourage users to be satisfied with recommendations that hold low utility.

In the first section of this paper we cover essential background knowledge in the stable marriage problem through set theory, graph theory, and network science. The results of these analyses are used to construct a model for an optimized online dating service in the second section. In the third section we compare the model with Tinder's operations to determine how premium features and other minor methods encourage de-optimization. We conclude by summarizing our findings and suggesting areas of future work.

2 Background

2.1 Stable Marriage Problem

The *stable marriage problem* is concerned with assigning pairings of partners between two distinct sets A and B. The assignment of pairs is said to be a *stable marriage* if there exists no elements $a \in A$ and $b \in B$ such that a and b are not paired together yet would prefer each other over their current partners [7]. This concept is demonstrated in Figure 1. The *Gale and Shapely algorithm* can be used to find a stable marriage between the sets A and B, provided that they are disjoint [7].

Consider a generic online dating service and assume that the collection of this service's users, its *user base*, can be divided into two disjoint sets A and B. These users will cease further use of the service if assigned into a stable marriage as they will be unable to find a partner they consider better. Assuming the rate at which the user base grows through new users is lower than the rate at which users leave the service, the service will soon go out of business.

Realistically, the conditions for the Gale and Shapely algorithm are not met in Tinder. Foremost, the two sets of partners must be disjoint [7] to guarantee the existence of a stable marriage. Traditionally these sets are assumed to be composed of heterosexual men and women respectively,

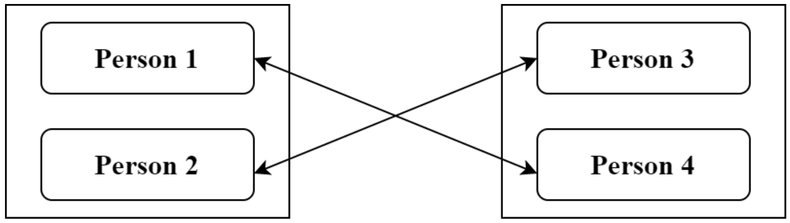


Figure 1: An example of expressed preferences between two disjoint sets. In this case, Person 1 and Person 4 prefer each other and Person 2 and Person 3 prefer each other. Pairing Person 1 with Person 4 and Person 2 with Person 3 results in a stable marriage. Any other assignment would result in an unstable marriage.

but this is not the case for Tinder where gender preference can be set regardless of the user's own gender [3]. That is to say, if Tinder's user base were divided into two sets, there is no guarantee that a member of one set would not prefer another member of the same set. Additionally, the stable marriage problem typically requires that members of one set rank the members of the other set in order of preference and vice versa [7]. Tinder does not allow ranking, instead opting for a simple approve/disapprove system. These shortcomings require the stable marriage problem to be applied within the context of graph theory.

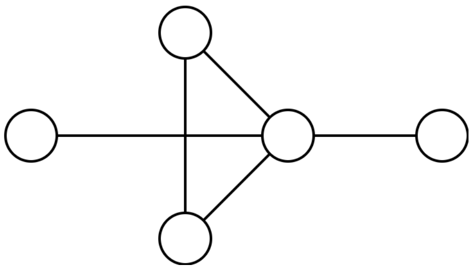
2.2 Graph Theory

In graph theory, a *graph* is defined as containing both a set of *vertices* and a set of *edges* disjoint from each other, with the restriction that the set of vertices is non-empty, as well as an *incidence function* which associates an unordered pair of vertices with each edge [2]. A graph G can be represented as

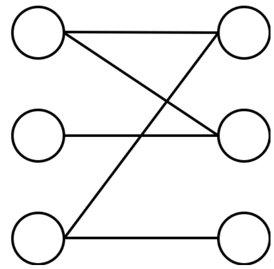
$$G = (V(G), E(G), \psi_G),$$

where $V(G)$ is the set of vertices, $E(G)$ is the set of edges, and ψ_G is the incidence function [2]. A *bipartite graph* is a specific type of graph wherein the vertex set can be partitioned into two subsets X and Y such that every edge in the edge set has one end in X and the other in Y [2], i.e. no vertices in the same set X or Y can be connected by an edge. Examples are shown in Figure 2.

A graph is considered k-regular if each vertex in the graph is adjacent to exactly k other vertices. This can also be referred to as each vertex having a *degree* of k and denoted as $d_G(v) = k$ for all $v \in V(G)$. A *perfect matching* is a set of edges in a graph such that each vertex in the graph is incident to an edge from the perfect matching, but no two edges in the perfect matching are incident with the same vertex. This concept is demonstrated in Figure 3. If a graph is both k-regular and bipartite for some $k > 0$, it must contain a perfect matching [2].



(a) A simple, non-bipartite graph.



(b) A simple, bipartite graph.

Figure 2: Two examples of a graph structure where vertices are connected by edges. The circles represent vertices and the lines between circles represent edges. Note that the bipartite graph can be represented by a partition of the vertices into two subsets such that there are no edges between vertices in the same subset.

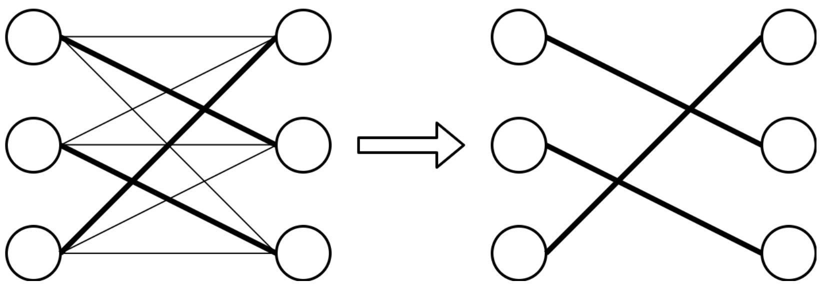


Figure 3: An example of a perfect matching in a 3-regular bipartite graph where no member of the matching is incident with the same vertex as another member and all vertices in the graph are incident with at least one member of the matching. Members of the perfect matching are represented by bold lines.

Consider a graph where each vertex represents a person and the edges between two vertices represent an expression of interest between two people. Note that an expression of interest is not a pairing. Additionally, let this graph be both k-regular and bipartite for some $k > 0$. Then there must exist a perfect matching within this graph. This allows us to pair each member of one partition with a member of the other partition in a manner analogous to a stable marriage [2]. Thus, we can consider finding a perfect matching within a k-regular, bipartite graph to be a solution to the stable marriage problem in the instance where each expressed interest is equally weighted.

Whereas set theory is inadequate for applying the stable marriage to Tinder, graph theory is sufficient for doing so. Let T be a bipartite graph representative of Tinder. Let the set X be the entirety of Tinder's user base such that each vertex represents a single user and let the set Y be a copy of the same user base. Then let T be a bipartite graph with the vertex set $X \cup Y$. Note that each user is then represented by two vertices in T. Let the edges between vertices represent a reciprocal expression of interest between users, i.e a *match* as it is referred to by Tinder. For now,

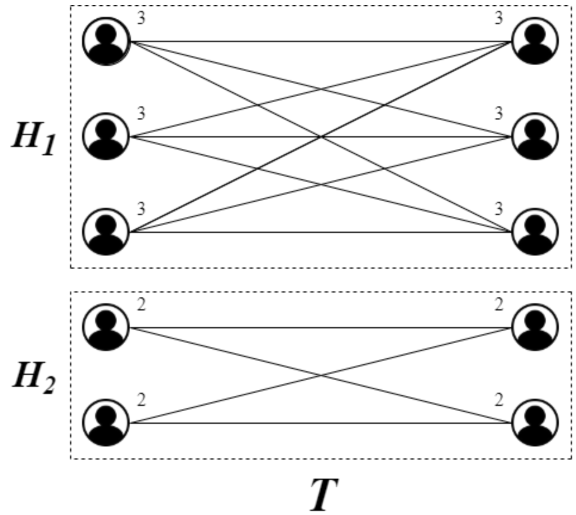


Figure 4: A bipartite graph T decomposed into two subgraphs H_1 and H_2 such that each subgraph is regular and H_1 and H_2 are both disjoint and edge-disjoint from each other. The degree of a vertex, where a vertex represents a user and an edge between vertices represents an expression of interest, is represented by the number adjacent to the vertex.

we will assume that a user has equal preference between all matches. If T could be shown to be k-regular (also referred to as just *regular*), then it could be stated that a stable marriage exists inside of Tinder. However, it may not be necessary for T to be regular in order to guarantee the existence of a stable marriage.

A graph H is defined as a subgraph of a graph G if $V(H) \subseteq V(G)$, $E(H) \subseteq E(G)$, and ψ_H is the restriction of ψ_G to $E(H)$ [2]. Additionally, we can consider two subgraphs H_1 and H_2 to be disjoint if they share no vertices and edge-disjoint if there are no edges that connect the vertices from H_1 to the vertices of H_2 and vice versa [2]. If it can be shown that T can be decomposed into the two subgraphs H_1 and H_2 and that both H_1 and H_2 satisfy the stable marriage problem, then it would be shown that T still satisfies the stable marriage problem even if T itself is not regular. An example of this decomposition can be seen in Figure 4. Note however that T does not necessarily have to be decomposed into only two subgraphs. This notion is extended into the following theorem.

Theorem 2.1. *Let G be composed of bipartite subgraphs that are both disjoint and edge-disjoint such that for sets X and Y of any subgraph the members of X have ranked only the members of Y in order of preference and vice versa. Assume a lack of preference is equivalent to the worst possible preference. Then the assignment of members in G is a stable marriage if the assignment of members in each subgraph is a stable marriage.*

Proof. Let H_α and H_β be two distinct subgraphs of G such that all members of a subgraph have been paired with another member of the same subgraph. Then, without loss of generality, for any $\alpha \in H_\alpha$ there is no element in H_β that α will prefer to its current assignment. This implies that there exists no $g_1, g_2 \in G$ such that both g_1 and g_2 would rather be assigned to each other than their current partner. Thus, the assignment of members in G is a stable marriage. $\square$

It is difficult, if not impossible, to determine if Tinder satisfies the conditions for Theorem 2.1. However, we argue that if enough regular subgraphs of sufficient size exist within Tinder's representative graph T, then consequently enough stable marriages may exist to greatly diminish Tinder's user base. After all, the formation of regular subgraphs is not unlikely. Tinder tailors a user's experience to their own personal preferences, such that the user will generally not encounter people who fall outside these preferences [3]. This greatly limits the recommendation pool for the user and, thus, it is likely that users with similar preferences will cluster together into smaller subgraphs or *communities*. If these users all match with each other the community would then become regular when observed independently of the larger graph.

We note that there is no guarantee that these communities are disjoint or edge-disjoint from each other. Indeed, considering that a person can belong to multiple communities, the small world effect [8] would seem to argue that any two communities are connected by a chain of an average length of six people. This is a seemingly non-existent distance compared to the millions of people that use Tinder. However, a lack of distinction may not be enough to completely throw out Theorem 2.1. For example, if all members of a community prefer each other over members of another community and vice versa, the two communities are equivalently disjoint, as the members' top interests are contained within a single community, and edge-disjoint, as removing the edges between these two subgraphs would have no effect on a stable marriage algorithm. Thus, a high count of regular communities in T could pose a potential problem for Tinder. This gives us the following corollary.

Corollary 2.1.1. *A graph which can be partitioned into regular subgraphs that are disjoint but not necessarily edge-disjoint may still satisfy Theorem 2.1 if the edges between two subgraphs represent a preference weaker than all preferences in both subgraphs.*

This corollary will be relied on heavily in later sections. However, one large issue remains with this model. It was assumed that users will have equal preference for each other, but this is a generous assumption that does not follow reality. While graph theory does possess the tools to overcome this assumption, concepts from network science will provide a more complete model.

2.3 Network Science

A *network* can be thought of as an applied form of a graph [8] and is generally considered to contain larger sets of vertices and edges in which patterns cannot be easily distinguishable by eye

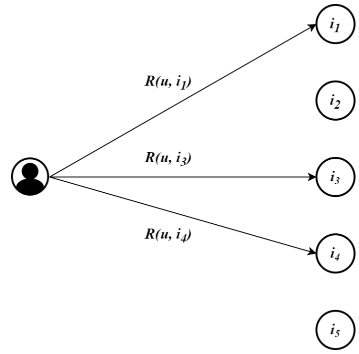


Figure 5: A preference network for a single user represented as a bipartite graph with directed, weighted edges. The left partition holds a single user and the right partition holds items that could be potentially recommended. The edges represent items that will be recommended to the user and the weight of the edges, denoted by the function $R(u, i)$, represents the value of the recommendation [8]. If an item is not connected to the user by an edge, then it will not be recommended.

[8]. Networks often make use of further graph concepts such as *directed edges*, where an edge is associated with a direction [8], and *edge weights*, where edges hold some form of value referred to as a *weight* [2]. Like graphs, networks can be classified into different types [8]. For example, a *preference network* is a *bipartite network* (analogous to a bipartite graph) with one partition representing individuals and the second representing objects of interest. The edges between vertices would then denote that an individual has an interest in some object and be weighted according to that interest [8]. This is demonstrated in Figure 5.

Preference networks are the underlying structure behind *recommender systems* [8], which are algorithms that provide recommendations to users [1]. A *reciprocal recommender system* utilizes the preferences of both the user and the object of interest (typically another user) to determine if they should be recommended to each other [1]. For example, such a recommender would not recommend a man who prefers men to a woman who prefers men. In essence, different users have different *utilities* to each other, where utilities represent the value of a recommendation [1]. Given a user u and an item i (in this instance, another user), the true utility $R(u, i)$ can be estimated by $\hat{R}(u, i)$ [1]. Consider a recommendation system that will recommend k items (referred to as top-k recommendation system [1]) given a large set of items denoted

$$\{i_1, i_2, \ldots i_n\},$$

where $k \geq 1$, $n \geq 1$, and $k << n$. The system will calculate a set of estimated utilities

$$\{\hat{R}(u, i_1), \hat{R}(u, i_2), \ldots, \hat{R}(u, i_n)\},$$

and then return the top k items with the highest estimated utility. This process is represented in Figure 6.

Figure 6: A representation of a top-k recommender system. The input is a set of n items and the output is a subset of the input containing the top k items to recommend.

If we assume Tinder provides users with recommendations that are meaningful to both the user's goal of forming a relationship and Tinder's goal of maintaining a user base, then $\hat{R}(u, i)$ must factor in the utility of the recommendation to both the two users as well as Tinder itself. For example, a user who will receive a lot of matches but is unlikely to form a relationship with any would have high utility for Tinder. They are not forming relationships with any users, so all users remain in the user base. However, as all users are receiving matches, they are likely to feel more satisfied with the service than if they were receiving no matches at all.

Returning to the preference network, the edge directed from User A to User B would now be weighted by $\hat{R}(A, B)$. In this sense, the edge now holds the direction of preference as well as the ranking of preference. Assuming that Tinder accurately estimates utility relative to the user's own preference, this concept can be used to create a model for Tinder without assuming that users hold equal preferences over each other. However, we will first use this concept to create a model for a generic optimal online dating service instead.

3 Model

Consider an optimal online dating service represented by the preference network

$$P = (V(P), E(P), \psi_P).$$

Let $V_X(P)$ and $V_Y(P)$ represent the vertices contained in sets X and Y respectively such that each vertex represents a user in the service and a single set contains a vertex for every user. Note then that for all vertices $x \in V_X(P)$ there exists a $y \in V_Y(P)$ such that both x and y represent the same user and vice versa. However, we will not consider these vertices to be equal, i.e. $x \neq y$, as we will let $V_X(P)$ represent the set of all individuals in P and $V_Y(P)$ represent the set of all objects of interest in P. This gives us that

$$V(P) = V_X(P) \cup V_Y(P).$$

Let an edge $e \in E(P)$ be a directed, weighted edge. The edge e represents a possible recommendation to be given to a user such that an edge directed from user $u \in V(P)$ to the user $w \in V(P)$ represents that w could be recommended to u. Note that this does not necessarily mean that w has been recommended to u yet, but rather that a recommendation is possible.

The weight of the edge will represent the estimated utility of the recommendation $\hat{R}(u, w)$. We assume that if u will be recommended to w then w will be recommended to u as well. However, the estimated utility of the recommendation of w to u may differ from the estimated utility of the recommendation of u to w. Essentially, we can assume that it will generally be the case that

$$\hat{R}(u, w) \neq \hat{R}(w, u),$$

for all $u, w \in V(P)$. Note that if $\hat{R}(u, w)$ or $\hat{R}(w, u)$ equal zero then neither user should be recommended to the other.

The incidence function ψ_P, which associates each edge $e \in E(P)$ with a pair of vertices $v_1, v_2 \in V(P)$, maintains its traditional definition. As such, given an edge $e \in E(P)$ that is directed from user $u \in V(P)$ to user $w \in V(P)$, we can define $\psi_P(e)$ as

$$\psi_P(e) = uw.$$

Corollary 2.1.1 states that a decomposition of P into a set of regular disjoint subgraphs H may satisfy Theorem 2.1 even if the subgraphs are not edge-disjoint. To this effect we will construct H such that the number of edges within each subgraph $h \in H$ is maximized while the number of edges between each subgraph is minimized. Additionally, all vertices of P will be represented by the set of subgraphs H, i.e. for all $v_P \in V(P)$ there exists an $h \in H$ and a $v_h \in V(h)$ such that $v_P = v_h$. An example of this decomposition process is demonstrated in Figure 7.

We assume that this service wishes to successfully create as many relationships as possible without any regard to maintaining profitability. In other words, we seek to form as many stable marriages as possible within the subgraphs of H. This can be accomplished through finding a perfect matching in P, which requires finding a perfect matching for all $h \in H$. Let

$$M_H = \{M_1, M_2, \ldots M_k\},$$

be the set of perfect matchings for $h_1, h_2, \ldots, h_k \in H$ respectively with $k = |H|$. Let these edges be picked dependent of their weights such that the stable marriage problem is satisifed for each subgraph h. Then the matching

$$M_P = M_1 \cup M_2 \cup \ldots \cup M_k,$$

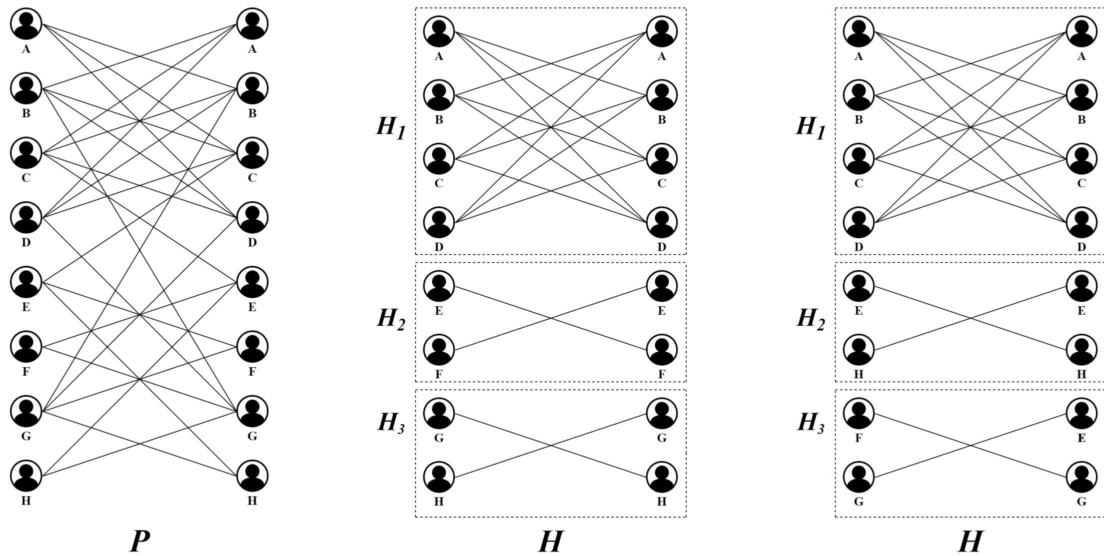


Figure 7: The preference network P can be decomposed into a set of disjoint subgraphs H such that each subgraph $h \in H$ is both regular and disjoint and the edges within subgraphs are maximized while the edges between subgraphs are minimized. Note that P may have multiple qualifying decompositions.

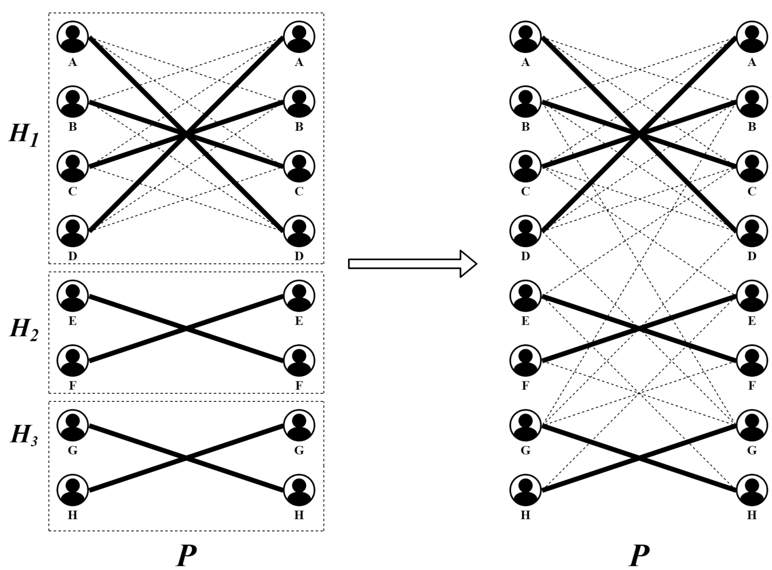


Figure 8: We can find a minimum maximal matching, which is also a perfect matching, for each $h \in H$. This represents a perfect matching in P.

can be considered a perfect matching for P as every vertex in P is incident to an edge in M_P and no two edges in M_P are incident to the same vertex in P.

The perfect matching of a graph G may be constructed while satisfying the stable marriage problem by finding a *minimum maximal matching* (MMM) for G [4]. Thus, if we let M_H be the set

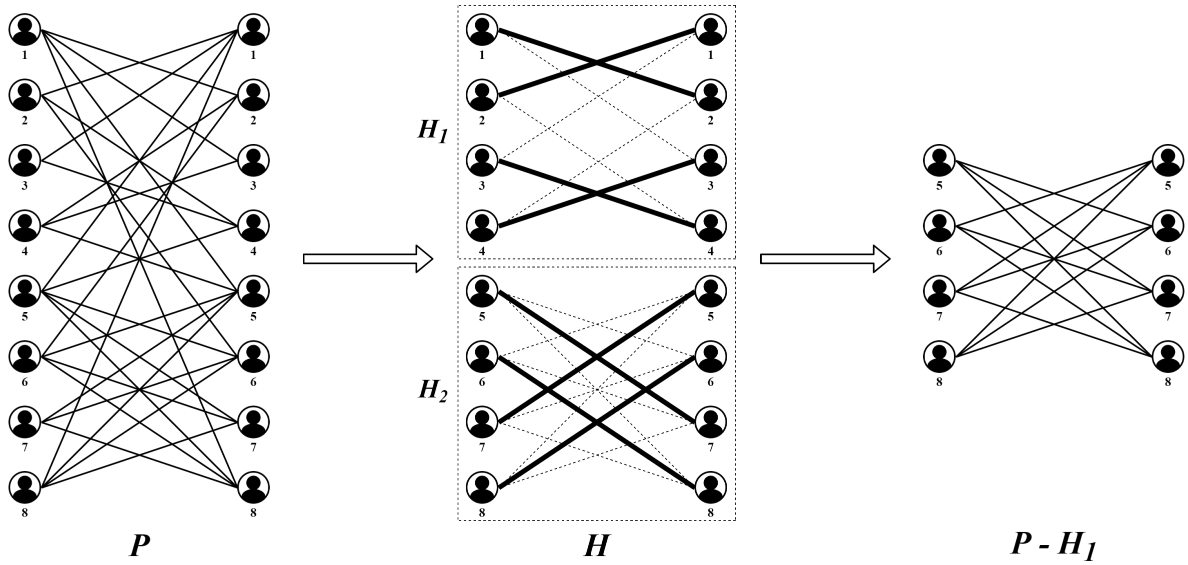


Figure 9: The set P decomposed into the set of subgraphs $H = \{H_1, H_2\}$. If the service's algorithm succeeds in creating stable marriages in only H_1, resulting in all users of H_1 leaving the service, we are left with $P - H_1$. The algorithm will execute again on $P - H_1$ and continue until the user base falls below the minimum viable population, at which point the user base will soon collapse to zero.

of MMM for each subgraph $h \in H$, we can arrive at the same conclusion that M_P will be a perfect matching which satisfies the stable marriage problem for P. This methodology is demonstrated in Figure 8.

Finally, assume that the service requires at least a certain amount of users to remain in business and assume that users assigned within a stable marriage are no longer considered part of the service's user base. This amount will be referred to as the *minimum viable threshold* and finds its inspiration in work on population models.

If we calculate the error between the estimated utility $\hat{R}$ determined by the service's recommender algorithm and the actual utility R, it is likely we would find that $\hat{R}$ may be allowed a large margin of error compared to R as the recommender algorithm would only need to create enough stable marriages to guarantee the user base drops below the minimum viable threshold. Thus, we can determine that the recommender algorithm used by the service does not need to be perfect. This concept is illustrated in Figure 9.

To summarize, the algorithm utilized by the optimal online dating service will operate as such. Begin by calculating $\hat{R}(u, w)$ for all $u \in V_X(P)$ and all $w \in V_Y(P)$ such that u and w do not represent the same user. Decompose P into a set of subgraphs H such that each subgraph in H is regular and disjoint and the number of edges within a subgraph is maximized while the number of edges between any two subgraphs of H is minimized. For each subgraph $h \in H$, compute the minimum maximal matching of h and let the set M_H represent the set of all minimum maximal matchings in H. Create a perfect matching for P by defining M_P as the union of all matchings contained in the set M_H. If two users are incident to an edge contained in M_P, recommend them to each other. If users u and w form a relationship, remove them from the user base. Repeat this process until the user base falls below the minimum viable threshold.

4 Results

It does not take much to show that Tinder could meet the conditions for an optimal online dating service. Consider the preference network P with the set of users $V(P)$ and the weighted, directed set of edges $E(P)$. Now let every user in Tinder's user base correspond to two v for every $v \in V(P)$; one vertex for each partition. Additionally, let every $e \in E(P)$ be directed according to the possible recommendation between two users and weighted by the estimated utility determined by Tinder. Tinder allows users to specify their personal preferences, which enables the formation of communities. This lets P be decomposed into the set of regular, disjoint subgraphs H that can be used to satisfy the stable marriage problem. Thus, Tinder can be considered an optimal online dating service if it recommends users based on perfect matchings formed in the members of H. However, as per the paradox discussed previously, Tinder cannot operate both optimally and as a successful business. Therefore, there must be some form of de-optimization that allows Tinder to balance optimization with business revenue.

We can define de-optimization as any act which would prevent or discourage the formation of a stable marriage within a community. Two features of Tinder that can be considered acts of de-optimization are limitation and recycling [10]. Limitation corresponds to limiting the amount of recommendations given to a user each day (unless the user pays a premium fee), whereas recycling refers to delivering recommendations that were previously rejected, i.e. delivering recommendations with a low estimated utility. Limitation can be considered a *weak* form of de-optimization as, in an optimal online dating service, it only delays the time taken for the user to be matched within a stable marriage. Recycling can be considered weak as well as it not only serves to delay similar to limitation, but also if a user's preferences change it is possible they can now be matched within a stable marriage to the recycled recommendation.

It was noted that limitation can be overcome through a premium fee. Therefore, even though it is weak, it is still a form of de-optimization that generates revenue and helps Tinder succeed as a business. In fact, we will find that *strong* forms of de-optimization are those which maximize revenue and minimize loss of user satisfaction. The strongest de-optimizations are then likely to be found in Tinder's suite of *premium features* that can be accessed by paying additional fees.

Passport is one example of a premium feature. Passport is described as allowing users to receive recommendations from any location, regardless of where the user themselves are located [11]. In essence, the feature lets the user's preference for location be taken out of the calculation of the estimated utility. With less input data, the estimated utility will become less accurate. Top Picks, which recommends more desirable users [13], is another example of strong de-optimization. Tinder has been known to calculate a desirability score for its users [10], but Top Picks will deliver recommendations that fall outside of this score, i.e. the feature removes another parameter for calculating estimated utility.

Tinder provides additional premium features beyond these two, but these two are sufficient for showing Tinder's approach to de-optimization. Both serve to strongly affect the estimated utility of a recommendation in an adverse manner. When paired together, especially with weaker forms of de-optimzation, the effect becomes stronger. Both features also generate additional revenue for Tinder, as they require a cost to be used. However, users will receive more interesting recommendations and will want to use the service longer in order to score a match, even if the match is now less likely due to a lower estimated utility. In essence, these features discourage the formation of stable marriages while both keeping users in the user base and generating revenue from said users.

While more difficult to prove, the possibility that Tinder may not deliver ideal recommendations should also be considered. Given the set of possible recommendations

$$I = \{i_1, i_2, \ldots, i_k\},$$

with k being an integer greater than or equal to one, Tinder need only supply a subset $\hat{I} \subseteq I$ of recommendations to the user such that $\hat{I}$ is composed of recommendations with lower utility. Alternatively, if the utility estimator $\hat{R}$ factors in the utility the users pose to Tinder, the elements of $\hat{I}$ could be chosen such that if two users were matched then their utility to Tinder would be minimally affected. For example, users who match through the Passport premium feature but are unlikely to form a relationship would still generate a high estimated utility.

It should also be noted that Tinder should not discourage the formation of stable marriages too highly as doing so would risk users becoming dissatisfied with the service. A previous study has found that browsing interesting profiles and receiving matches is positively correlated with *satisfaction* with Tinder [3]. Additionally, it has been found that running out of profiles to swipe on is positively correlated with *dissatisfaction* [3]. The second point can be remedied with premium subscriptions that grant additional or unlimited swipe access, but the first point is the fulcrum with which Tinder must balance its algorithm. If only recommendations with low estimated utility are provided, users will become dissatisfied. However, if only recommendations with high utility are provided, users will become satisfied too quickly. All the meanwhile, Tinder must be able to maintain a steady stream of revenue.

5 Conclusion

This paper began with the paradox that Tinder cannot perform optimally without failing as a business. The stable marriage problem was applied to set theory, graph theory and network science to determine how an optimal online dating service would function. A model for the optimal online dating service was then constructed. We found that Tinder could be an optimal online dating service, but it intentionally de-optimizes itself through methods such as alternative recommendation delivery methods or, more likely, premium features.

Strong de-optimization methods such as the premium features Passport or Top Picks seem ideal for Tinder to maintain the balance between user satisfaction and business success. This paper attempts to determine possible locations for de-optimization in an optimal online dating service, but remains within the abstract for the most part. Future work may find benefits in determining how likely users are to continue using Tinder after experiencing de-optimization, especially through methods presented in premium features.

One generous assumption this paper makes is that users of Tinder are looking for long-term relationships. It is possible that users may instead be seeking friendships, short-term commitments, or some other form of interaction (e.g. just to chat). However, these alternate use cases are unlikely to impact Tinder's de-optimization methods. For example, while users who are seeking friendships are unlikely to leave the service after forming a single friendship, they are still likely to leave the service after forming enough friends as there are only so many active friendships they can sustain. As another example, a user seeking a short-term relationship may find themselves unintentionally in a long-term relationship as the result of Tinder's optimization. In either case, it is to Tinder's benefit to continue to employ de-optimization methods, whether weak or strong.

It is also possible that Tinder does not employ any de-optimization methods at all and that it is the users themselves who are responsible for the de-optimization, e.g by not messaging matches or not looking at profiles long enough to determine if they hold any appeal. This seems unlikely, and it could be argued that such methods are actually de-optimizations encouraged by Tinder, but this may be another area future work could look into.

Overall, it is not difficult to construct an overview of Tinder's algorithms, even if those algorithms remain proprietary and confidential. The question is how Tinder adjusts the parameters of these algorithms, and the answer to this question must explain how the paradox that an optimal online dating service cannot exist as a business is resolved. This paper constructs the overview and asks the question, but ultimately suggests only one possible answer. Only those who know the algorithms can answer the question truthfully, but there remains much insight to be gained by coming to our own conclusions.

6 Acknowledgements

I would like to thank my mentor Dr. Mary Lynn Reed as well as Dr. Darren Narayan for their assistance and insight in the development of this paper.

References

[1] Eric Andrews. Recommender systems for online dating. Master's thesis, University of Helsinki, 2015.

[2] J.A. Bondy and U.S.R Murty. *Graph Theory With Applications*. Elsevier Science Publishing Co., Inc., 1976.

[3] Cédric Courtois and Elisabeth Timmermans. Cracking the tinder code: An experience sampling approach to the dynamics and impact of platform governing algorithms. *Journal of Computer-Mediated Communication*, 23:1–16, 2018.

[4] M. Demange and T. Ekim. Minimum maximal matching is np-hard in regular bipartite graphs. *Theory and Applications of Models of Computation*, 5:364–374, 2008.

[5] Mansoor Iqbal. Tinder revenue and usage statistics (2020), 2020. `https://www.businessofapps.com/data/tinder-statistics/` [Accessed: March 21, 2020].

[6] Paige Leskin. Dating apps like tinder, match, and bumble are still growing, but analysts predict that growth will 'slow significantly' in 2019, 2019. `https://www.businessinsider.com/dating-app-usage-growth-slowing-tinder-match-bumble-analysts-say-2019-6` [Accessed: March 29, 2020].

[7] D.G. McVitie and L.B. Wilson. The stable marriage problem. *Communications of the ACM*, 14(7):486 – 490, 1971.

[8] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003.

[9] Gwendolyn Seidman. Is tinder really a hookup app?, 2017. `https://www.psychologytoday.com/us/blog/close-encounters/201706/is-tinder-really-hookup-app` [Accessed: March 21, 2020].

[10] Kaitlyn Tiffany. The tinder algorithm, explained, 2019. `https://www.vox.com/2019/2/7/18210998/tinder-algorithm-swiping-tips-dating-app-science` [Accessed: April 13, 2020].

[11] Tinder.com. Passport, 2020. https://www.help.tinder.com/hc/en-us/articles/115004490423-Passport [Accessed: April 16, 2020].

[12] Tinder.com. Tinder plus and tinder gold, 2020. https://www.help.tinder.com/hc/en-us/articles/115004487406-Tinder-Plus-and-Tinder-Gold- [Accessed: March 30, 2020].

[13] Tinder.com. Top picks, 2020. https://www.help.tinder.com/hc/en-us/articles/360005039092-Top-Picks [Accessed: April 16, 2020].