附录3：（封面、封底用浅蓝色无花纹封面纸（卡纸）打印，无须彩打。此面为封面正面。所有红字在打印之前必须删除。）



# 本科生毕业设计（论文）参考文献译文本

|  |  |
| --- | --- |
| 译文出处 | **IEEE Transactions on Multi-Scale Computing Systems ( Volume: 4, Issue: 3, July-Sept. 1 2018)** |
| 院 系 | **计算机科学与技术学院** |
| 专业班级 | **计算机科学与技术（国际交流）201701班** |
| 姓 名 | **俞景昳** |
| 学 号 | **U201713732** |
| 指导教师 | **华强胜** |

2021 年 3 月

**译文要求**（封面、封底用浅蓝色封面纸（卡纸）打印，此面为封面背面）

1. 译文内容须与课题（或专业内容）联系，并需在封面注明详细出处。
2. 出处格式为

图书：作者.书名.版本（第×版）.译者.出版地：出版者，出版年.起页～止页  
期刊：作者.文章名称.期刊名称，年号，卷号（期号）：起页～止页

1. 译文不少于5000汉字（或2万印刷符）。
2. 翻译内容用五号宋体字编辑，采用A4号纸双面打印，封面与封底采用浅蓝色封面纸（卡纸）打印。要求内容明确，语句通顺。
3. 译文及其相应参考文献一起装订，顺序依次为封面、译文、文献。
4. 翻译应在第七学期完成。

**译文评阅**

|  |
| --- |
| **导师评语**  应根据学校“译文要求”，对学生译文翻译的准确性、翻译数量以及译文的文字表述情况等做具体的评价后，再评分。 |
| 评分：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_（百分制） 指导教师（签名）：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  年 月 日 |
|  |

**Incremental Maintenance of Maximal Bicliques in a Dynamic Bipartite Graph**

**摘要**

我们考虑了一个随着新增边变化的动态二部图中最大二分团的增长维护。当新的变被添加进图中时，我们试图去枚举最大二分团的变化，而不是枚举不受影响的最大二分团的集合。这个有效算法的挑战就是在不显式地枚举最大二分团的集合。在这篇工作中，我们给出了(1)由于边集的变化而引起的图的最大二叉集变化量的近紧界，以及(2)列举最大二叉集变化量的增量算法。对于在图中添加常数条边的情况，我们的算法是“变化敏感的”，也就是说，它的时间复杂度与最大二分团的变化量成正比。据我们所知，这是第一个在动态图中枚举最大双点的增量算法，具有可证明的性能保证。我们的算法易于实现，实验结果表明，它的性能比基准实现高出几个数量级。

**关键词—— 图挖掘，最大二分团维护，增量算法，二部图**

1. **介绍**

在许多领域，如社会网络分析、计算机生物学和网络搜索中，图是表示关联数据的一种自然抽象。通常，这些网络是动态的，新的连接被添加，旧的连接被删除。动态图挖掘领域的重点是在动态图中寻找和维护重要模式的有效方法。我们的工作是由需要从动态图中维护稠密子结构的应用中获得启发的。Angel等人[6]，提出了一种基于在线社交网络中信息中实体的共生关系的算法，通过从进化图中挖掘稠密子图来实时识别突发新闻故事。Java等人的[17]提出了一种检测微博平台用户间社区的方法，该方法通过识别代表用户间连接的演化网络中的稠密结构来实现。其他应用程序的示例包括，稠密子图挖掘在网络包括识别社区社交网络[15]、[23],识别网络社区[14]、[34]、[18],种系发生树构造[11]、[35]、[44]，二部图中的社群[19]，基因组分析[30],和闭项集挖掘[41]、[20]。

我们将重点放在可以用来为两种不同类型实体之间的交互建模的二部网络上。例如，一个提要中的用户和新闻文章之间的关系可以被建模为二部网络，其中用户和新闻文章是两个顶点集，如果用户已经看过一篇文章，则用户和一篇新闻文章之间存在一条边。我们研究了演化二部图中的最大二分团枚举（MBE）。MBE是一个基本的问题，已被用于检测社区在大型二部网络。例如，Kumar等人的工作，从网络图中检测在线社区[18]，Murata从网络日志数据中识别用户社区[29]，和Lehmann等人在协作网络社区检测[19]，都是基于一个适当定义的二部图中的MBE。MBE的另一个应用是从事务性数据库中挖掘关联规则挖掘中的封闭项集[31]。一个方法就是，从一个表示事物数据库的二部图中枚举最大二分团，其中一部分顶点时不同的事务，另一部分的顶点时事项集，如果一个事项被包含在一个事务中，那么两个顶点间存在一条边[20]。

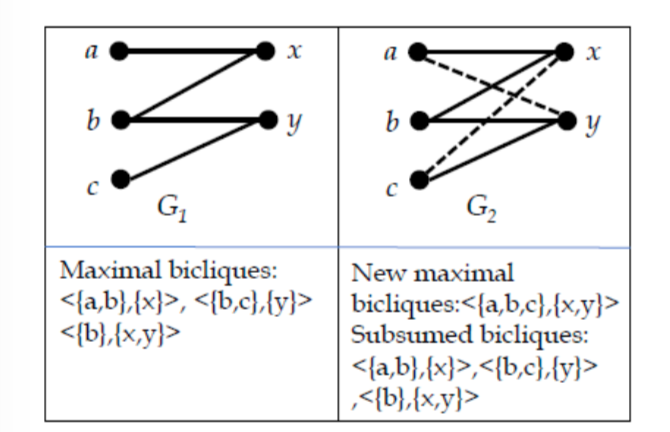
以前关于MBE的大部分工作都集中在静态图上，这种静态图假定是完整给出的，而且今后不会改变。这样的方法效率不高，而且通常不适用于图形经常变化的情况，比如向数据库添加更多事务，或者网络中有更多的用户活动。例如，在社区检测中，如果将边添加到图中，那么就不清楚如何有效地(重新)计算出现的新社区的出现，以及被吸纳的社区。在事项集挖掘的场景下，当事务以增量方式添加到数据库时，(重新)计算关闭的事项集是一个很大挑战[41]。当今的许多，如果不是大多数的话，数据源都是动态的，并不断生成数据，因此有一个能够以有效方式处理此类动态数据的方法是很重要的。

我们考虑了由于新边的增加而不断演化的二部图中的最大的二部图维护，也就是增量MBE问题。设为一个简单的无向二部图，其顶点集划分为，和边集合。一个中的二部图是一个二分，当X中的每一个顶点连接到Y中的的每一个顶点。一个二分B被称作最大二分团，如果不存在一个二分团，使得是其真子集。我们让表示G中的最大二分团集合。

假设我们从二部图开始考虑，其状态因为新的边集合H的加入，变为。我们让表示在中出现的新最大二分团，而原来不在中出现的；我们让表示愿阿里在中出现的最大二分团，而现在不在中出现的。让表示他们的对称差。我们考虑下面两个问题：

(1)的大小可以为多大，尤其是，一个小的边集改变能否导致最大二分团集合的巨大改变？

(2)我们如何去高效地计算？我们可以很快地计算出当很小的时候吗？简单地说，我们可以设计一个变化敏感算法，来去枚举的元素，它的时间复杂度和的大小成比例？



* 1. **贡献**

**改变的幅度：**

让表示在一个n顶点的二部图中，最大二分团的最大数量。Prisner[33]的结论表明，，当n为偶数的时候等号成立。我们证明当单一的边加入图的时候，最大二分团的变化可以大到，这指数比于图的顶点。这个表明，即使是一条边加入图中，也可以导致最大二分团集合的巨大变化。我们之后证明了这个界限的是紧确的，当只有一条边加入，最大可能的变化为。

**枚举算法：**

通过我们的分析，最大二分团的变化在最坏的情况下和n成指数关系。在一方面

这个变化可以小到1。举个例子，考虑一个新的到达的边连接两个孤立顶点的情况。因此，最大二分团的变化的变化范围是非常大的。当幅度很大的时候，算法枚举所有的变化一定是很耗时的，如果仅仅只去枚举变化。再另一方面，当变化的幅度很小的时候，它讲理想地用很少的开销。这个引发了我们对研究计算开销和最大二分团变化幅度成线性比例的算法的兴趣。

我们证明了一个增量算法，DynamicBC，当有一组边被加入二部图G的时候，去枚举最大二分团的变化。DynamicBC有2个部分，NewBC，去美剧新的最大二分团；SubBC，去枚举那些已经被吸纳了的最大二分团。当有一个批次的新边，其大小为，NewBC时间复杂度为，其中为图更新后最大的度。SubBC的时间复杂度为。注意到当为一个常数的时候，时间复杂度线性比于所输出的二分团，然后乘上一个和图大小有关的系数。据我们所致，这是第一个动态图中维护最大二分团的变化敏感的算法。

**实验评价：**

我们给出了DynamicBC在具有百万节点的真实二部图上的经验评价，并将我们的算法与基准方法方法进行了比较。我们的结果表明，DynamicBC的性能比直接应用静态算法(BaselineBC)快许多个数量级，比我们设计的改进基线(BaselineBC\*)快许多倍。例如，在lastfm-song-init图上，DynamicBC花了大约93秒来计算由于添加了625批每个大小为100的批所造成的更改，而BaselineBC则花了大约7,920秒，BaselineBC\*则花了大约1,740秒。

1. **前言**

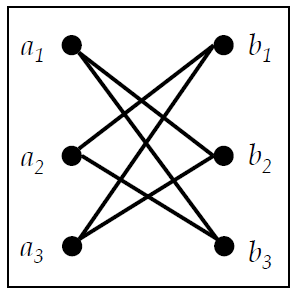
设为的顶点集，为的边集，设和分别为的顶点数和边数。让表示中与顶点相邻的顶点集。如果从上下文来看，图G的含义是清晰的，我们使用表示。对于一条边，让表示从G删除后的图,表示图添加为一组边到G后的图。对于一组边集，让表示添加(删除)到（从）之后的图。同理，对于顶点，设表示将加到后的图，对于顶点，设表示从中删除及其所有相邻边后的图。设为中顶点的最大度，为中顶点的最小度。

**定义1(变化敏感算法)**。一个针对动态图中属性P的算法是变化敏感的，当它枚举P的变化的的时间复杂度是和P变化幅度呈线性关系，并且和输入图的大小、边集的大小呈多项式关系。

我们注意到，这种“变化敏感”的用法与文献中“输出敏感”一词的用法是一致的。从静态图中枚举最大二角的算法被称为“输出敏感”，如果运行时间与输出的大小(最大二分团的总数)呈线性关系，再乘以图中顶点数和边数的因子多项式。例如，Alexe等人[5]提出的MBE的输出敏感算法需要的时间，其中n是图中顶点的个数，β是最大二分团的个数，这是相关的输出。

**静态图的结果。**在[33]中，Prisner给出了下列的关于最大二分团的结果。让表示鸡尾酒图，其是每一个部分有着k个顶点的一个二部图。和。

**定理1(定理2.1 [33])** 每一个有n个顶点二部图最多包含个最大二分团，并且极大二分团图仅是。



作为一个子过程，我们使用了一种从静态无向图中枚举最大二分团的算法，其运行时间与最大二分团的个数成正比。有一些这样的算法[5]，[22]，[45]。我们使用Liu等人的[22]得出的以下结果，因为它提供了当前的最佳时间复杂度。

**定理2(Liu等人 [22])** 对于一个包含了n个顶点，m条边，且最大的度为，最大二分团的数量为的图G，存在一个MinLMBC的算法去枚举最大的二分团。其时间复杂度为，空间复杂度为。

MineLMBC是一个静态图枚举最大二分团的算法，其基于DFS。它将图G和阈值s作为输入，这个算法会枚举每个部分大小最少为s的全部的最大二分团。很明显，如果s=1，那么这个算法将会枚举全部的最大二分团。

* 1. **动机**

应用程序可以比文件系统更好地了解它们写入的数据的生存期和更新频率。但是，应用程序不知道文件系统元数据的生存期和更新频率。文件系统元数据的更新频率通常与应用程序的数据不同，并且经常受到特定文件系统的磁盘布局和写策略的影响。通常，文件系统在逻辑上保持数据和元数据的分离，但它们在固态硬盘上可能不会保持“物理上”的分离。在执行文件操作时，元数据写操作可能会与同一NAND块中的数据写操作混合，或者一种元数据可能会与另一种元数据混合。具有流分离能力的文件系统可以减少应用程序数据和文件系统元数据的混合，提高写放大系数和性能。

1. **最大二分团的算法**

对于一个图，和边集合，我们使用去表示，同时用去表示。在展示我们的变化敏感算法之前，我们先考虑2个基准方法。

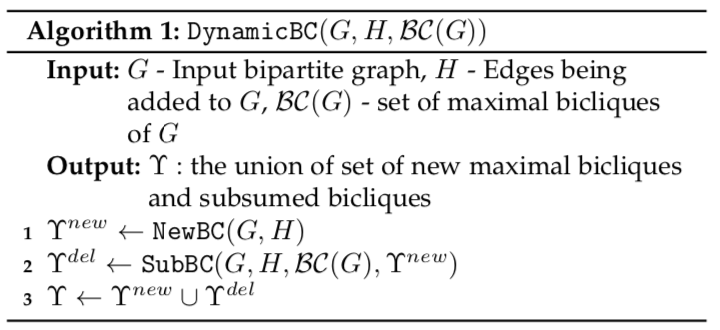
* 1. **Baseline算法**

首先，我们考虑一种直接的方法，使用静态图中最棒的算法来维护最大二分团。该算法称为BaselineBC，它的工作原理是枚举，一旦用一组新的边更新了，那么在中所有最大二分团的集合也会更新，然后输出(保存在内存中)和之间的对称差。

接下来，我们呈现另一个基准算法BaselineBC\*，它比BaselineBC好。它的想法是关注变化发生的部分。让表示所有G中顶点的集合，且至少连接了H中的一条边。为了去枚举新的最大二分团，我们主要到考虑即可，它是通过得到的。BaselineBC\*使用静态图中最先进的算法去枚举所有最大二分团。每个由此生成的二分团b是一个最大的二分团，当其包含中至少一条边。为了去枚举被吸纳的二分团，我们注意到每一个被吸纳的二分团是至少一个最大二分团的子图，并且也一定被包含。因此，被吸纳的最大二分团通过考虑每一个新的最大二分团b和b-H从而被枚举。如果一个最大二分团因此被枚举，且同时在中，那么它就被输出为一个被吸纳的二分团。

我们可以期望BaselineBC\*比BaselineBC做得更好。BaselineBC\*它仍然是不变化敏感的，因为它可能在枚举新的最大二分团的过程中，重复生成中的最大二分团。接下来，我们给出了小心地避免枚举重复生成中的最大二分团算法。

* 1. **变化敏感算法 DynamicBC**

****

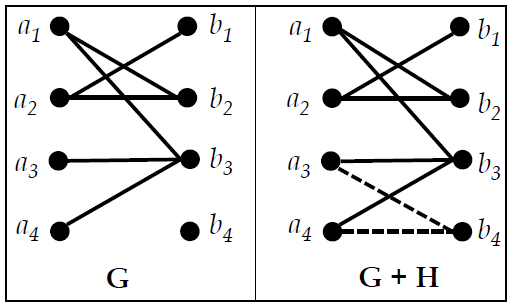
我们的变化敏感算法，DynamicBC，有两个部分，1）算法NewBC去生成新的最大二二分团，在3.3节描述；2）算法SubBC去枚举被吸纳的二分团，在3.4节描述。主要的时间结果关于时间复杂度将由下面的定理给出。

**定理3.** DynamicBC枚举了最大二分团的变化，其时间复杂度为，其中为图的最大的度，是新增边集合的大小。

我们注意到，如果是一个常数的话，那么枚举的时间复杂度就为。因为有以下的发现。

**发现1** 对于MBE问题而言，DynamicBC是一个变化敏感的算法，当是一个常数。

* 1. **枚举新的最大二分团**

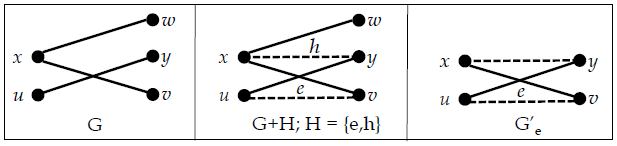


在我们的算法中，我们需要每一个被NewBC枚举的最大的二分团需要包含至少一条来自H的边，因此强制它成为一个新的最大二分团。我们让表示。对每一条新的边，让表示中包含边e的最大二分团。

**引理1.**

证明. 对于每一个在中的二分团，一定包含至少一条中的边。为了去证明这个，我们考虑一个二分团，，如果不包含中的一条边，那么也是一个中的最大二分团，因此不能属于。因此对于某个边，也就得到了成立。这个证明了。然后我们再来考虑对于某个二分团，一定存在的情况，对于中的某个成立。因此b是一个中的最大二分团。因为包含了边，所以b不能是中的一个二分团。因此.这也就证明了。

接下来，对于每一条边，我们展示一个高效的方式去枚举所有中的二分团，通过枚举某一个特殊子图的最大二分的方式来获得。由下面的方式构造。让,。然后就是一个生成子图了。可以看下图。

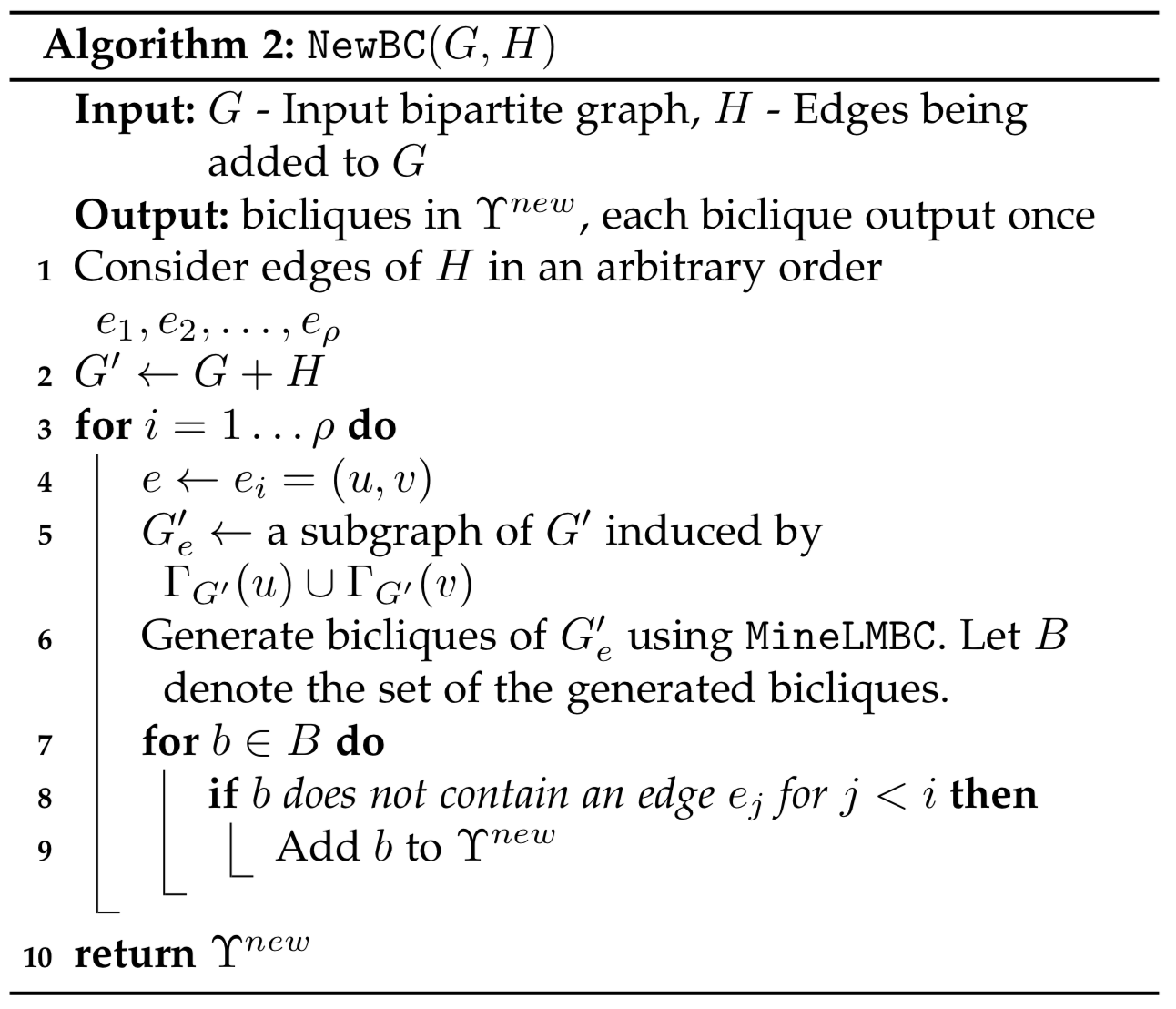


**引理1. 对于每一个**

**证明.** 首先，我们证明。考虑一个二分团。让很明显，包含了边因为它包含了顶点和.假设。通过构造时候的限制，我们知道所有顶点都和相连接。然后在中，所有在中的顶点都连接到了中的顶点。因此是一个中的二分团。并且，是一个中的最大的二分团，同时是的一个生成的子图，包含了所有中的顶点。因此就是一个的最大二分团。

接下来，我们证明。考虑一个在中的二分团。很明显，包含边因为它同时包含了和，并且是一个的最大二分团。因此，还是一个包含边的二分团。现在我们来证明也是一个最大二分团。假设不是，那么这里就有一个顶点且可以通过扩展。然后对于每一个构造生成的，，因为一定要和连接。所以，就不是一个中的最大的二分图。这就矛盾了，因此也是一个的最大二分图。因此， 。

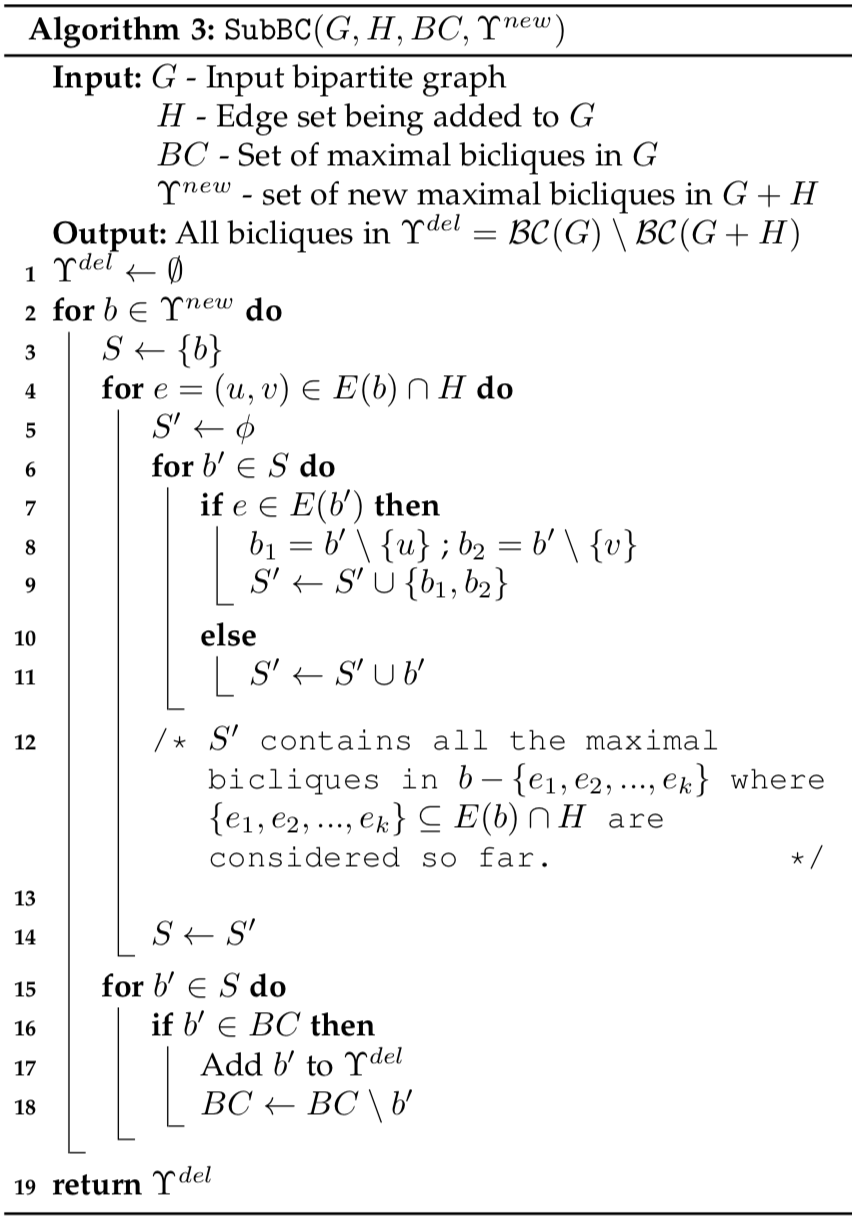
基于上述的观察，我们展示我们的改变敏感算法NewBC（算法2）。我们使用一个用在静态图中的输出敏感算法MineLMBC去枚举最大的二分团。注意到，通常来说，比小得很多，因为其是基于边的，因此枚举所有的最大二分子图的开销会相对来说不昂贵。



**定理4**. NewBC 枚举所有的因为新边集合增加的二分团集合，耗时在，其中为图的最大的度，为的大小。空间复杂度为。

**证明.**首先，我们考虑这个算法的正确性。通过引理1和引理2，我们知道了是通过枚举每一条边的来获得的。我们的算法精确地做到了这个，并且使用MineLMBC算法去枚举。在运行时，考虑到这个算法每次迭代都是关于每一条边。在每一次迭代，它构造了一个图，然后运行MineLMBC(。注意到顶点的数量是不会超过2的，因为这个大小为中条的一条的邻居的数量的集合。每一次迭代生成的最大二分团是的一个子集，因此每一次迭代生成的数量不会超过。从定理2中，我们知道了每一次迭代的运行时间为，因为这里有条边在中，运行的结果会在下面给出。对于空间复杂度，我们注意到算法在任何时间都不存储新的二分团于内存中。构造的空间是线性的，然后大小为。通过定理2，全部需要的空间是。

* 1. **枚举被吸纳的最大二分团**

****

我们现在考虑枚举。假设一个新的最大二分团吸收了一个中的最大二分团注意到也是一个在中的最大的二分团。一个方法是要去枚举所有最大二分团，然后检查他们是否在中。但是，检查二分团是一种非常耗时的操作，因为我们需要考虑二分团的每一个顶点的相邻顶点。另一个想法是去显式存储图的二分团，并且去观察它是否被包含。这也是不理想的，因为这需要很大的存储空间去存储这么多的二分团。

我们考虑了一个更加有效地方法，去将这些最大二分团的特征储存，而不是这些最大二分团本身。我们然后去枚举所有中的最大二分团，然后对于每一个因此产生的二分团，我们去比较它们的特征和已经存储的特征。一个这样想法的算法在算法3中展示了。这减小了内存的消耗。我们使用了一个标准的哈希函数（64位 murmur哈希）。为了去计算一个二分团的特征，首先我们表示一个二分团通过标准行事（第一个集合里面顶点的字典序，紧跟着第二个集合里面顶点的字典许）。然后我们将这个字符串转换为字节，并且应用哈希函数去生成特征。通过存储哈希特征而不是最大二分团的本身，我们有能力去通过比较它们的哈希值，从而迅速的检查中的最大二分团是否已经被中包含了。和存储整个二分团相比较，我们同样花了很少的内存代价。

这个存储二分团特征的方法，又很小的概率会产生哈希碰撞，也就是说两个二分团A和B不一样，但是它们的哈希值是一样的。这个碰撞影响是误检——我们的算法可能会错误的推断一个二分团是一个被吸纳的二分团，但实际上并不是。但是，这是一个非常不可能发生的时间，通过使用64位特征，两个不同的字符串有同样的哈希值的几率是非常小的。在我们的实验中，我们算法枚举出来的二分团的集合总是匹配得上被吸纳的二分团。注意到我们可以总是二次检查每一个二分团，通过显示的检查它是否是中的最大团，去避免误检的可能性。我们没有在我们的实现中实现它，因为这个几率太小了。

现在我们来证明算法3枚举了所有中的的最大二分团。

**引理3.** 在算法3中，对于每一个，14行之后的S包含了中所有最大的二分团。

**证明.** 首先我们观察到，从中移除和从中移除在中出现的的边。因此计算的最大二分团可以简化为计算的最大二分团，其中是中所有出现在中的边的集合。我们从的边数入手，考虑基本的情况，包含了一条边。很明显，有两个最大二分团和。假设集合有大小，我们的推导的假设是的全部二分团都被枚举了。考虑有条边的。现在每一个最大二分团或者在中保持最大（至少有一个的顶点不在中）或者生成了两个的最大二分团（如果两个端点都在）中。因此，对于每一个，14行之后的包含了所有的的最大二分团。

现在我们证明上述的算法是一个对于枚举所有是变化敏感的算法，当中边的数量是一个常数。

**定理5.** 算法3枚举了中全部的二分团，，时间开销为，其中为为边集H的数量。算法的空间开销为。

**证明.** 我们首先证明每一个被该算法枚举的二分团b’确实是y中的一个二分团。注意到b’是一个G中的二分团，因为算法中的显式条件检查。此外，b’并不是G+H中的一个最大二分团，因为它是b的一个真子图，一个G+H的最大二分团。接来下，我们证明所有y的二分团都被枚举了。考虑任意一个被吸纳的二分团b’。它一定被b\H包含，其中b是y的一个最大二分团。此外通过引理3可知，b’也将是b\H的一个最大二分团，且将会被该算法枚举。

对于时间复杂度，从p归纳，我们证明了对于任意b，b-H中最大的二分团数量是2p。假设p=1，那么H包含了一条边，我们称之为e1。那么b-H有2个最大二部图，b\{u}，这就证明了基础情况。假设对于任意一个大小为k的集合H，那么b-H将不会有超过2k个最大二部图。考虑到一个有k+1条边的集合H’’。让H’，子图b-H’’可以从b-H’删除一条边e获得。可以推导出，b-H’将不会有超过2K的最大二部图。每一个最大二部图或者在xx中仍为最大二部图（如果至少一个顶点不再b’）,或者至少得到两个最大二部图（如果xx在b两部分）。因此，b-H的最大二部图的数量将不会超过2k+1，我们就完成了数学归纳法。基于此，对于任意一个二部图b，我们需要去检查其最大型对于不超过2p个G的二部图。这个可以通过检查每一个生成的二部图是否被包含在了集合BC中，对于每一个二部图，这个操作可以在常数时间内完成。

对于空间限制，

1. **实验评价**
   1. **数据**
   2. **实验设置和实现**
   3. **结果讨论**
2. **总结**

**参考文献**（宋体5号加粗）

[1]□王静康,张凤宝,夏淑倩等.论化工本科专业国际认证与国内认证的“实质性”.高等工程教育研究,2014,5:1-4

[2]□Stone J A, Howard L P. A simple technique for observing periodic nonlinearities in Michelson interferometers. Precision Engineering,1998,22(4):220-232

[3]□朱印红,袁衍明.Dreamweaver完美网页设计——技术入门篇.(第一版).北京:中国电力出版社,2006:19～20

[4]□Lewis S L. Physics and chemistry of the solar system.北京:北京大学出版社,2014.1～2

[5]□陈剑.上博简《民之父母》“而得既塞於四海矣”句解释[EB/OL］.简帛研究网站，http://www.bamboosilk.org/Wssf/2003/chenjian03.htm．2003-01-18

( 宋体5号)

（可加页，A4纸双面打印）

**参考文献原文**