****

**الجمهورية العربية السورية**

**وزارة التعليم العالي**

**الجامعة الافتراضية السورية**

**ماجستير علوم الويب**

**تقرير رسالة الماجستير**

**للشهر (4+5)**

|  |  |
| --- | --- |
| **اسم المشرف** | **د. باسل الخطيب** |
| **بريد المشرف** | [**t\_balkhatib@svuonline.org**](mailto:t_balkhatib@svuonline.org) |
| **اسم الطالب** | **م.علي عسكر** |
| **بريد الطالب** | [**ali\_133380@svuonline.org**](mailto:ali_133380@svuonline.org) |
| **عنوان الرسالة بالعربية** | **تحسين سرعة حل مسألة المواءمة العظمى عبر تحسين توليد مرشح أولي بالاعتماد على الحدس** |
| **عنوان الرسالة بالانكليزية** | **Optimize solution runtime of maximum Bipartite matching problem by optimizing the generated initial candidate based on heuristic** |
| **تاريخ التسليم** | **28-05-2023** |

**خطوات الإنجاز في هذا التقرير**

**تمت في هذه المرحلة دراسة وبرمجة**

* خوارزميات إيجاد التدفق و الخوارزميات المقادة بالحدس لإيجاد المطابقة من حيث الآلية و زمن التنفيذ على بيانات ثنائية عشوائية كثيفة وغير كثيفة أيضا بناء خوارزمية حدس كما ورد في الخطة وقياس ادائها مقارنة بنظيراتها:
* **خوارزميات إيجاد التدفق:**

1. خوارزمية Ford-Fulkerson
2. خوارزمية Edmond-Karp
3. خوارزمية Dinic’s

* **خوارزميات الحدس:**

1. خوارزمية Monte-Carlo
2. خوارزمية Simple-Greedy
3. خوارزمية Randomized-Rounding
4. خوارزمية Dynamic-Min-Degree
5. خوارزمية Static-Min-Degree
6. خوارزمية Limit-Min-Degree
7. الخوارزمية التراجعية Backtracking

* **خوارزميات إيجاد التدفق**
* **خوارزمية Ford-Fulkerson و Edmond-Karp**

حيث تتشابه الخوارزميتان في كل شيء والاختلاف فقط في خوارزمية إيجاد المسار المتبقي حيث تعتمد خوارزمية

Ford-Fulkerson على خوارزمية البحث في العمق أما Edmond-Karp ف تقوم على خوارزمية البحث في العرض

12

20

16

10

4

9

7

13

4

14

12/0

20/0

16/0

7/0

9/0

4/0

10/0

13/0

4/0

14/0

**Total-Flow = 0**

**بتطبيق خوارزمية إيجاد المسار (DFS or BFS) نجد 3 مسارات و هي:**

1. **0—1—3—5**

**ونختار أقل وزن على طول المسار وهو 12 ونضيفها إلى التدفق لتصبح الشبكة المتبقية على الشكل التالي**

**Total-flow = Total-flow + 12 = 12**

20/12

12/12

12/16

7/0

10/0

4/0

9/0

13/0

4/0

14/0

1. **0—2—4—5**

**ونختار أقل وزن على طول المسار وهو 4 ونضيفها إلى التدفق لتصبح الشبكة المتبقية على الشكل التالي**

**Total-flow = Total-flow + 4 = 16**

12/12

20/12

12/16

9/0

10/0

4/0

7/0

4/4

4/13

4/14

1. **0—2—4—3—5**

**ونختار أقل وزن على طول المسار وهو 7ونضيفها إلى التدفق لتصبح الشبكة المتبقية على الشكل التالي**

**Total-flow = Total-flow + 7 = 23**

12/12

20/19

12/16

4/0

10/0

7/7

9/0

11/14

4/4

11/13

**لا يوجد مسارات أخرى بين العقدة 0-5 وبالتالي التدفق الأعظمي 23**

* **خوارزمية Dinic’s**

وتقوم على مبدأ منع التدفق و إرسال التدفق في دفعات أي أنه يتم تحديد المسارات بين العقدة والمصب بحيث كل عقدة تنتمي إلى المسار يجب أن تكون إلى مستوى أقل ب واحد من العقدة التي تليها حيث يتم تحديد المستويات بتطبيق خوارزمية البحث في العرض و ومن ثم إيجاد المسارات التي تحقق المعادلة السابقة باستخدام خوارزمية البحث في العمق وإرسال تدفق في جميع المسارات التي تم إيجادها سابقا.

4

10

10

8

6

2

10

10

9

**Total-flow = 0**

**في التكرار الأول:**

نقوم بإسناد المستويات لكل العقد باستخدام خوارزمية البحث في العرض ونقوم بفحص البيان للتأكد فيما إذا كان هنا المزيد من التدفق الممكن إي مسار من ال عقدة 0 إلى العقدة 5 في البيان المتبقي

Level on node

2

1

4

0

10

10

6

8

3

2

10

10

9

2

1

Cross edge can’t be used to send more flow in this iteration because these edges don’t connect node from 1 th to i+1 th level

**Total flow = Total flow + 4 + 6 + 4 = 14**

الأن نستطيع منع التدفق باستخدم المستويات أي أنه كل مسار للتدفق يجب أن يحور على المتسويات التالية 0,1,2,3 ونقوم بإرسال التدفقات المتاحة معا وهنا يكمن التحسين الذي طرق على خوارزمية Edmond-Karp حيث نقوم بإرسال تدفق واحد في التكرار الواحد

**4 على المسار 0-1-3-5**

**6 على المسار 0-1-4-5**

**4 على المسار 0-2-4-5**

التدفق الكلي = التدفق الكلي + 4 + 6 +4

في التكرار الأول تكون الشبكة المتبقية على الشكل التالي

4

4

10

6

2

6

2

6

6

4

5

10

4

**في التكرار الثاني:**

نقوم بإسناد المستويات الجديدة لكل العقد باستخدام خوارزمية البحث في العرض على البيان المتبقي السابق ونقوم بالتحقق فيما إذا كان هناك المزيد من التدفق ( أي مسار بين العقدة 0 و العقدة 5)

Those edges that are not go from i th level node to (i+1) th level node are crossed

3

3

4

10

4

4

0

6

2

6

2

6

5

10

6

4

4

2

1

الأن تستطيع منع التدفق باستخدام المستويات والذي يعني كل مسار يجب أن يحوي على المستويات

1,2,3,4 لهذا نستطيع إرسال التدفق فقط عبر المسار 0-2-4-3-5 بقيمة 5 ليصبح التدفق الكامل  
Total-Flow = Total-flow + 5 = 19 و تكون الشبكة المتبقية على الشكل التالي:

4

9

10

1

2

5

1

2

6

1

10

9

9

**في التكرار الثالث:**

نقوم بتطبيق خوارزمية البحث في العرض لإيجاد مستويات العقد ونتأكد فيما اذا كان يوجد مسار بين المنبع والمصب وتتوقف الخوارزمية إذا كان لا يوجد مسار أخر بين المنبع والمصب في البيان المتبقي وهنا نلاحظ لا يوجد مسار ف تتوقف الخوارزمية ويكون التدفق الأعظمي هو 19

**تاريخ خوارزميات إيجاد التدفق**

* **خوارزميةFord-Fulkerson**

و هي خوارزمية كلاسيكية تستخدم للعثور على التدفق الأعظمي في شبكة ما تم اقتراحها لأول مرة من الدكتور فورد الابن " L. R. Ford Jr" والدكتور فولكرسون " D. R. Fulkerson” عام 1956 تستخدم الخوارزمية مفهوم الشبكة المتبقية للعثور بشكل متكرر على المسار المتبقي والتي تستخدم لزيادة التدفق في الشبكة حتى الوصول للحد الأقصى

تعمل الخوارزمية من خلال البدء بتدفق أولي بقيمة 0 ثم البحث بشكل متكرر على مسار لزيادة التدفق وتتكرر هذه العملية حتى تختفي المسارات حيث لا يوجد طريق بين العقدة والمصب

يمكن أيضا أطلاق طريقة بدلا من خوارمية حيث لم يتم تحديد طريقة إيجاد المسار المتبقي بالكامل.

* **خوارزمية Edmons-Karp**

وهي أحد أشكال خوارزمية فورد-فولكرسون لإيجاد أقصى تدفق تم تطويرها من قبل العالم

جاد إدموندز " Jack Edmonds "وريتشارد كارب " Richard Karp" في عام 1972 تستخدم الخوارزمية البحث في العرض للعثور على المسارات المتبقية في شبكات التدفق مما يؤدي إلى تحسين الطرق التقليدية في إيجاد المسارات مقارنة بخوارزمية فورد فولكيرسون أي أن ادموند-كارب هي تخصيص لخوارزمية فورد-فولكيرسون.

* **خوارزمية Dinic’s**

خوارزمية دينيك وتعد تباينا لخوازرمية فورد-فولكيرسون لإيجاد التدفق الأعظمي تم تطويرها من قبل العالم السوفييتي ييفيم دينيك "Yefim A. Dinic" وتستند الخوارزمية إلى مفهوم البيانات الطبقية في بعض حالات البيان تكون الخوارزمية أسرع من سابقاتها   
يمكن أن يختلف أداء خوارزميات ford-Fulkerson و Edmonds-Karp و Dinic اعتمادا على الخصائص المحددة لشبكة التدفق التي يتم تحليلها

بشكل عام تتمتع خوارزمية فورد-فولكيرسون بتعقيد زمني أسوأ من الخوارزميات الأخرى ولكنها يمكن أن تكون الأفضل على بعض أنواع البيانات وضمن التنفيذات العملية للخوارزمية في أغلب الحالات وذلك سيتم إيضاحه في جداول التنفيذ على بيانات عشوائية كثيفة وغير كثيفة

تم بناء البيانات العشوائية بحالة خاصة (بيانات ثنائية) وإضاة عقدة و مصب وتوجيه البيان ليصبح البيان مناسب ك دخل للخوارزميات السابقة وبالتالي التدفق الأعظمي يساوي إلى المطابقة القصوى حيث وزن جميع الوصلات 1

تم توليد البيانات عبر توليد جميع الوصلات الممكنة وتخزينها في بنية معطيات و تطبيق توابع خلط وأخذ قسم معين من الوصلات بناء على درجة الكثافة المطلوبة وفق المعادلة التالية

عدد الوصلات = (عدد عقد القسم الأول من البيان \* عدد عقد القسم الثاني من البيان) \* درجة الكثافة

والنتيجة يتم تدويرها إلى عدد صحيح.

* **خوارزميات الحدس في إيجاد المطابقة**
* **خوارزمية Monte-Carlo**

تعد خوارزمية مونت كارلو لتحقيق أقصى قدر من المطابقة خوارزمية إرشادية عشوائية تستخدم الاحتمال لإيجاد حل تقريبي لمشكلة المطابقة القصوى في الرسم البياني. غالبًا ما تُستخدم هذه الخوارزمية في المواقف التي يكون فيها إيجاد حل دقيق باستخدام خوارزمية مثل Edmonds-Karp أو Hopcroft-Karp مكلفًا أو غير عملي من الناحية الحسابية.

تعمل خوارزمية مونت كارلو عن طريق تحديد الحواف عشوائيًا من الرسم البياني وإضافتها إلى مجموعة مطابقة ، طالما أنها لا تنتهك قيود المطابقة. تتكرر هذه العملية عدة مرات ، ويتم تحديد مجموعة المطابقة التي تنتج أكبر عدد من الحواف على أنها الحد الأقصى التقريبي للمطابقة.

عدد التكرارات k هو عامل يؤثر على دقة الخوارزمية.

تؤدي القيمة الأكبر لـ k عادةً إلى تقريب أكثر دقة للحد الأقصى للمطابقة ، ولكنها تزيد أيضًا من التكلفة الحسابية للخوارزمية. تعد خوارزمية مونت كارلو خوارزمية إرشادية ، مما يعني أنها لا تضمن الحل الأمثل.

ومع ذلك ، يمكن أن يكون نهجًا فعالًا لإيجاد حلول تقريبية للرسوم البيانية الكبيرة حيث تكون الخوارزميات الدقيقة باهظة التكلفة أو غير عملية من الناحية الحسابية.

يمكن تحسين دقة الخوارزمية عن طريق زيادة عدد التكرارات .التعقيد الحسابي هو O(k \* n^2)

* **خوارزمية Simple-Greedy**  
  تعد خوارزمية Simple-Greedy طريقة إرشادية لإيجاد أقصى تطابق في رسم بياني ثنائي الأجزاء. تحدد الخوارزمية بشكل متكرر الحواف المتوافقة مع المطابقة الحالية حتى لا يمكن إضافة المزيد من الحواف. تحتوي الخوارزمية على تعقيد زمني لـ O (mn) ، حيث m هو عدد الحواف و n هو عدد العقد في الرسم البياني ثنائي الأجزاء. ومع ذلك ، فإن خوارزمية Simple-Greedy ليست مضمونة للعثور على أقصى تطابق في جميع الحالات. في الواقع ، هناك العديد من الأمثلة حيث فشلت الخوارزمية في العثور على الحد الأقصى من المطابقة.
* **خوارزمية Randomized-Rounding**  
    
   الاستدلال العشوائي لأقصى قدر من المطابقة هو خوارزمية تستخدم العشوائية للعثور على تقريب جيد لمشكلة المطابقة القصوى في البيانات الثنائية. يُطلق على هذه الخوارزمية اسم الكشف عن مجريات الأمور لأنها لا تضمن حلاً مثاليًا ، بل هي حل شبه أمثل باحتمالية معينة. يعمل الاسترشاد العشوائي لأقصى قدر من المطابقة على النحو التالي: ابدأ بمطابقة فارغة M. لكل حافة (u ، v) في الرسم البياني الثنائي ، اخترها بشكل مستقل لتكون في M مع احتمال 1/2. عودة M. الفكرة الرئيسية وراء هذه الخوارزمية هي أنه من خلال التحديد العشوائي للحواف مع احتمال 1/2 ، يمكننا إنشاء مطابقة قريبة من الحد الأقصى للمطابقة. احتمال اختيار أي حافة معطاة هو 1/2 ، وبالتالي فإن الحجم المتوقع للمطابقة هو نصف حجم الحد الأقصى للمطابقة تقريبًا. من خلال تشغيل الخوارزمية عدة مرات واختيار أفضل تطابق تم العثور عليه ، يمكننا زيادة احتمال العثور على تطابق أكبر. تتمثل إحدى ميزات هذه الخوارزمية في بساطتها وسهولة تنفيذها. لا يتطلب سوى بضعة أسطر من التعليمات البرمجية ويمكن تطبيقه على الرسوم البيانية الكبيرة بملايين الحواف. بالإضافة إلى ذلك ، يمكن موازاة الخوارزمية وتوزيعها بسهولة ، مما يسمح لها بالاستفادة من بنيات الحوسبة المتوازية الحديثة.
* **خوارزمية Dynamic-Min-Degree**

يُعد الاستدلال الديناميكي من الدرجة الدنيا خوارزمية معروفة لإيجاد أقصى تطابق في رسم بياني ثنائي الأجزاء. تعتمد هذه الخوارزمية على فكرة اختيار العقدة دائمًا بأقل درجة ثم تقليل درجة العقد المجاورة.

يحتوي الاستدلال الديناميكي من الدرجة الدنيا على تعقيد زمني لـ O (m \* n ^ 2) ، حيث m هو عدد الحواف في الرسم البياني ثنائي الأجزاء و n هو عدد العقد في الرسم البياني. ثبت أن هذه الخوارزمية تؤدي أداءً أفضل من الأساليب التجريبية الأخرى في الممارسة ، خاصةً عندما يكون الرسم البياني متناثرًا. ومع ذلك ، فإنه لا توجد دائمًا أقصى تطابق في الرسم البياني.

* **خوارزمية Static-Min-Degree**

لاستدلال الاسترشادي من الدرجة الدنيا هو خوارزمية بسيطة لإيجاد أقصى تطابق في رسم بياني ثنائي الأجزاء. يعتمد على فكرة اختيار الرؤوس ذات الحد الأدنى من الدرجات (أي الحد الأدنى لعدد الحواف التي تقع عليها) بطريقة جشعة.  
الاستدلال على الدرجات الثابتة هي خوارزمية متعددة الحدود في الوقت O (m \* n) ، حيث m هو عدد الحواف و n هو عدد الرؤوس في الرسم البياني الثنائي الأجزاء.

* **خوارزمية Limit-Min-Degree**

بشكل مماثل لخوارزمية الدرجة الأقل الديناميكية لكن فقط نركز على العقد ذات الدرجات الأقل من قيمة معينة

يوجد العديد من التقنيات لحساب هذه القيمة أشهرها والتي تم استخدامها في برمجة الخوارزمية  
المدى بمعنى آخر (2 / القيمة الكبرى للدرجات – القيمة الصغرى)

تملك ذات درجة تعقيد الخوارزمية الديناميكية في أسوأ الحالات لكن في الحالات الواقعية توفر قدرا من الجهد بحيث تقوم بتخفيف مرات ترتيب العقد ذات الدرجة الأقل

بعد الانتهاء من العقد تقوم الخوارزمية بمطابقة العقد ذات الدرجات الأكبر من الثابت بشكل عشوائي.

* **الخوارزمية التراجعية Backtracking-Algorithm**

كما ورد في خطة البحث ف الهدف من البحث إيجاد تحسين على خوارزميات الحدس وقد تم ذلك عبر إيجاد خوارزمية والتي تعمل وفق المبدأ التالي

* أوجد الحد الأقصى للمطابقة عبر اختيار العدد الأصغر بين طرفي البيان الثنائي بالترتيب او وفقا للدرجة الاقل
* قم بالمرور على جميع عقد البيان في طرف ما r
* افحص جيران العقدةr وليكن n
  1. إذا n لم يتم مطابقته بعد قم بإضافة الوصلة n-r إلى مجموعة المطابقة
  2. إذ كان لا يوجد عقدة حرة من بين الجيران العقدة الحالية قم بالتالي
     + 1. ابحث عن بديل من بين مجموعة الوصلات المتطابقة بحيث يكون احد طرفاها هو n
       2. في حال وجدت هذه الوصلة-n uوإلا تجاهل التكرار الحالي وتابع للجار التالي
       3. افحص مجموعة الوصلات الغير متطابقة عن وصلة) u-v لتعويض التراجع عن اختيار الوصلة السابقة)
       4. في حال وجدت قم بإضافة كلتا الوصلتينr-n ,u-v للمطابقة واحذف الوصلة الموجودة أصلا في المطابقة **r-u**  وأعدها للوصلات الغير متطابقة
       5. اضف العقد الجديدة r,u من طرفي الوصلتين إلى مجموعة العقد المتطابقة.
* تابع التكرار حتى يصل عدد الوصلات المطابقة إلى الحد الأقصى للمطابقة او يتحقق المرور على جميع العقد

**البسودو-كود للخوارزمية التراجعية في إيجاد المطابقة**

* Given Bipartite Graph with (R, B, E) where R red nodes, B blude nodes, E edges
* Define: matched\_edges = {}, matched\_nodes = {}, un\_matched\_edges = {E}
* MAX\_VALUE\_MATCHING = min (length(R), length(B))
* For r in R:
  1. neighbors = neighbors (r)
  2. **if there is neighbor from neighbors not exists in matched\_nodes:**

1. add (node, neighbor) to matched\_edges
2. remove (node, neighbor) from un\_matched\_edges
3. add node to matched\_nodes
4. add neighbor to matched\_nodes
   1. **Else**:

* **for each n in neighbors:**

1. find u where u, n in matched\_edges
2. find (u, v) in un\_matched\_edges where v not in matched\_nodes
3. **if (u, v) exists:**
   1. remove (u, n) from matched\_edges
   2. add (u, v) to matched\_edges
   3. add (r, n) to matched\_edges
   4. add v to matched\_nodes
   5. add r to matched\_edges
   6. if length of matched edges = MAX\_VALUE\_MATCHING:

return matched\_edges

* Return matched\_edges

**مثال** :

تبدأ الخوارزمية بحساب الحد الأقصى للمطابقة وهو العدد الأصغر بين طويلة جهتي البيان وهنا الطرفان 4

وبالتالي الحد الأقصى للمطابقة هو 4

نقوم بتهيئة الوصلات التي لا تنتمي للمطابقة بجميع وصلات البيان و العقد التي تمت مطابقتها و الوصلات المتطابقة بمجموعات خالية :

un\_matched\_edges = {(A,1), (A,3), (A,4), (B,1), (C,3), (D,2)}

matched edges = {}

matched\_nodes = {}

نقوم بالمرور على عقد الجهة الاولى من البيان نبدأ من العقدة A نجد ان جيرانها 1,3,4 نختار عقدة عشوائية ولتكن1 و

من ثم نضيف الوصلة A-1 إلى مجموعة الوصلات المتطابقة ونقوم بحذف الوصلة من مجموعة الوصلات غير المتطابقة أيضا نضيف العقدة A والعقدة 1 إلى مجموعة العقد المتطابقة.  
لتصبح الحالة بنى التخزين بعد أول تكرار:

un\_matched\_edges = {(A,3), (A,4), (B,1), (C,3), (D,2)}

matched edges = {(A,1)}

matched\_nodes = {A, 1}

بالمرور على العقدة التالية B نجد أنها تملك جار وحيد وهو 1 ولكن تمت مطابقته وهنا تكمن فكرة التراجع في الخوارزمية

حيث تقوم الخوارزمية بفحص العقدة الجار للعقدة B والتي هي 1 في مثالنا هذا في مجموعة الوصلات المتطابقة لنجد انها تمت مطابقتها مع العقدة A وبذلك نقوم بفحص مجموعة الوصلات الغير متطابقة والتي أحد طرفاها هو العقدة A لإيجاد بديل بحيث نحافظ على مطابقة العقدة A ونوفر مساحة للعقدة A والعقدة 1 للمطابقة

لنجد ان الوصلة A-3 تحقق الشروط السابقة وبذلك نسبتدل الوصلة A-1 بالوصلة A-3 في مجموعة الوصلات التي تنتمي للمطابقة وإعادة الوصلة A-1 إلى مجموعة الوصلات الغير متطابقة

ونضيف أيضا ونضيف B-1 إلى الوصلات المتطابقة وأيضا نحذف B-1 من الوصلات غير المتطابقة

ونضيفها إلى الوصلات المتطابقة وايضا إلى مجموعة العقد المتطابقة نضيف العقدة B والعقدة 3.

لتصبح حالة بنى التخزين على الشكل التالي

un\_matched\_edges = {(A,1), (A,4), (C,3), (D,2)}

matched edges = {(A,3), (B,1)}

matched\_nodes = {A, B, 1, 3}

نقوم بالمرور على العقدة C لنجد انه يوجد جار وجيد وهو 3 نفحص العقدة 3 من الوصلات المتطابقة لنجد أنها تمت مطابقتها مع العقدة A

نقوم بفحص فيما إذا يوجد بديل لمطابقة A مع عقدة حرة ونجد في مجموعة الوصلات غير المتطابقة الوصلة A-4

سنقوم باستبدال الوصلة A-3 بالوصلة A-4 لتصبح العقدة 3 حرة ليتم مطابقتها مع العقدة C

وحذف الوصلة C-3 من مجموعة الوصلات الغير متطابقة وإضافتها إلى مجموعة المطابقة وإضافة العقدة 4 والعقدة C إلى مجموعة العقدة المتطابقة لتصبح حالة بنى التخزين على الشكل التالي

un\_matched\_edges = {(A,1), (A,4), (C,3), (D,2)}

matched edges = {(A,4), (B,1), (C,3)}

matched\_nodes = {A, B, C, 1, 3, 4}

نقوم بالمرور على العقدة D لنجد أنه يوجد جار لم يتم مطابقته وهو 2 لنقوم بإضافة الوصلة D,2 إلى مجموعة الوصلات المتطابقة لتصبح حالة بنى التخزين على الشكل التالي

un\_matched\_edges = {(A,1), (A,4), (C,3), (D,2)}

matched edges = {(A,4), (B,1), (C,3), (D,2)}

matched\_nodes = {A, B, C, D, 1, 2, 3, 4}

وبذلك يتحقق احد شروط توقف الخوارزمية في حالاتنا هنا تحقق كلا الشرطين

عدد وصلات المطابقة يساوي 4 وتم المرور على جميع عقد الجهة الأولى

لتكون المطابقة هنا هي **{(A,4), (B,1), (C,3), (D,2)} وهي مطابقة ولكنها قصوى في هذا المثال**

**إثبات عدم أمثلية الخوارزمية**

للوهلة الأولى قد تبدو الخوارزمية قادرة على إيجاد المطابقة القصوى ولكن فيما يلي مثال إثبات عدم أمثلية الخوارزمية في حالات معينة من توزع الوصلات والعقد نستطيع أن نجد حالات لعدم الأمثلية وهو السبب الذي يجعل الخوارزمية تعتمد ك خوارزمية مقادة بالحدس سريعة وفعالة في البيانات التي تنتمي لأرض الواقع وليس لأغراض أكاديمية حيث تقدم حلول قريبة إلى الحلول الأمثلية بدقة تتجاوز ال 90% و أكثر

مثال عدم الأمثلية:

بتفيذ الخوارزمية السابقة على البيان سنجد أن المطابقة يتم إيجادها بالتكرارات:

1. A🡪2
2. B🡪3
3. D🡪9
4. E🡪1
5. في هذا التكرار لن نجد عقدة حرة للمطابقة مع العقدى F وكما تنص الخوارزمية سنحاول إيجاد بديل للعقدة A لتحرير العقدة 1 والتي هي الجار الوحيد للعقدة F لنجد أنه لا يوجد أي عقدة حرة أو وصلة لا تنتمي للمطابقة تتحقق مطابقة العقدة A وتحرير العقدة 1 وبالتالي الخوارزمية ستجد أن المطابقة هي الوصلات السابقة

بينما المطابقة القصوى هي{(A,3), (B,5), (D,9), (E,1), (F,2)}

**وبالاستفادة من التجربة المقدمة من الخوارزميات المرتبطة بدرجة العقدة كما تم التوضيح سابقا نستطيع التخفيف من احتمال ورود هذا الالتباس بالمرور على عقد البيان بداية من درجة العقدة الأقل بدلا من المرور على عقد البيان بالترتيب.**

**دراسة زمن تنفيذ الخوارزميات والنتائج**

نستعرض في هذا الفصل مقارنات ونتائج لحساب زمن تنفيذ الخوارزميات السابقة على بيانات عشوائية البناء لملاحظة الفروقات الزمنية بشكل واضح قمنا بتطبيق الخوارزميات بحجم كبيرة نسبيا.

المقارنات تمت على بيانات ثنائية عشوائية كثيفة وغير كثيفة حيث:

وتم تمثيل النتائج في جداول ومخططات بيانية لتوضيح زمن تنفيذ الخوارزمية والفروقات الزمنية

التنفيذ تم على معالج وحيد بدون تفريع

**تم تحديد ذاكرة الوصول العشوائية ب 12GB و معالج بسرعة على حاسوبDell**

الحاسب المحمول يعمل بكفاءة أعلى على التيار الكهربائي عوضا عن البطارية بنسبة 20% كما نشرت لجنة الدفاع عن الموارد الطبيعية في الولايات المتحدة الأميركية في تقرير لها.

التنفيذ تم على تيار كهربائي لجميع التجارب لتوحيد مصدر الطاقة لجميع التجارب حيث جميع التجارب على نفس المعالج ونفس الظروف

تم بناء البيانات بناء على العشوائية حيث يتم توليد جميع الوصلات المحتملة بين طرفي البيان بناء على عدد العقد المعطى وتخزين هذه الوصلات في بنية معطيات وخلطها باستخدام بعض التوابع واخذ وصلات من هذه المجموعة بناء على درجة الكثافة للبيان حيث

**عدد الوصلات = عدد عقد الطرف الأول \* عدد عقد الطرف الثاني \* الكثافة**

بناء على العدد يتم أخذ الوصلات وهي أمثلة تطبيقية على بيانات واقعية والتي أثبتت الدراسات أن بعض الخوارزميات التي تبدو غير فعالة في الأوراق البحثية و الأكاديمية قد تكون أكثر فعالية على أرض الواقع وهو ما سيتبين معنا بالنتائج

في الختام ، يقدم هذا الفصل دراسة شاملة لوقت تنفيذ الخوارزميات المختلفة على البيانات التي تم إنشاؤها عشوائيًا. من خلال إجراء مقارنات دقيقة وتقديم نتائج مفصلة ، نقدم رؤى قيمة حول أداء هذه الخوارزميات في ظل ظروف مختلفة. يضمن تضمين العروض المرئية والاختيار الدقيق للأجهزة وإنشاء بيانات هادفة موثوقية

من خلال هذه الدراسة ، نهدف إلى المساهمة في التحسين المستمر وتطوير مناهج الخوارزميات ، مما يؤدي في النهاية إلى تعزيز قدرات صنع القرار وحل المشكلات في مختلف المجالات.

تم في البداية دراسة خوارزميات التدفق حيث تم دراسة الخوارزمياتDinic’s, Edmond-Karp, Ford-Fulkerson

من ثم دراسة الخوارزميات المقادة بالحدس

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| الاختصار | **G** | **N** | **D** | **MM** | **R** | **T** | **MC** | **SG** | **RR** | **SMD** | **DMM** | **LMM** | **BA** |
| الكلمة | **graph** | **Num**  **Of**  **nodes** | **density** | **Max**  **matching** | **result** | **Execution**  **time** | **Monte**  **Carlo** | **Simple**  **greedy** | **Randomized**  **Rounding** | **Static**  **Min**  **degree** | **Dynamic**  **Min**  **degree** | **Limit**  **Min**  **Degree** | **Backtracking**  **Algorithm** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Max Flow Value** | **Dinic’s Time** | **Edmond-Karp Time** | **Ford-Fulkerson Time** | **Density** | **Num of nodes** |
| 7 | 0.0029 | 0.002 | 0.003 | 0.0001 | 500 |
| 11 | 0.003 | 0.0026 | 0.002 | 0.0002 | 500 |
| 23 | 0.003 | 0.0029 | 0.003 | 0.0004 | 500 |
| 39 | 0.005 | 0.004 | 0.003 | 0.0007 | 500 |
| 49 | 0.005 | 0.005 | 0.004 | 0.001 | 500 |
| 117 | 0.021 | 0.019 | 0.009 | 0.003 | 500 |
| 160 | 0.05 | 0.038 | 0.02 | 0.005 | 500 |
| 198 | 0.07 | 0.052 | 0.02 | 0.008 | 500 |
| 223 | 0.08 | 0.0654 | 0.03 | 0.01 | 500 |
| 250 | 0.202 | 0.139 | 0.043 | 0.05 | 500 |
| 250 | 0.345 | 0.208 | 0.0589 | 0.1 | 500 |
| 250 | 0.517 | 0.331 | 0.091 | 0.18 | 500 |
| 250 | 0.696 | 0.408 | 0.118 | 0.25 | 500 |
| 250 | 0.967 | 0.587 | 0.164 | 0.36 | 500 |
| 250 | 1.14 | 0.665 | 0.191 | 0.42 | 500 |
| 250 | 1.282 | 0.771 | 0.226 | 0.48 | 500 |
| 250 | 1.515 | 0.871 | 0.253 | 0.56 | 500 |
| 250 | 1.643 | 0.962 | 0.279 | 0.62 | 500 |
| 250 | 1.874 | 1.118 | 0.303 | 0.68 | 500 |
| 250 | 2.01 | 1.147 | 0.34 | 0.72 | 500 |
| 250 | 2.13 | 1.237 | 0.362 | 0.78 | 500 |
| 250 | 2.22 | 1.287 | 0.369 | 0.82 | 500 |
| 250 | 2.378 | 1.413 | 0.396 | 0.88 | 500 |
| 250 | 2.793 | 1.6 | 0.460 | 0.92 | 500 |
| 250 | 3.024 | 1.69 | 0.515 | 0.98 | 500 |
| 250 | 3.064 | 1.77 | 0.529 | 1 | 500 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Max Flow Value** | **Dinic’s Time** | **Edmond-Karp Time** | **Ford-Fulkerson Time** | **Density** | **Num of nodes** |
| 24 | 0.006 | 0.006 | 0.006 | 0.0001 | 1000 |
| 45 | 0.009 | 0.0079 | 0.0069 | 0.0002 | 1000 |
| 90 | 0.017 | 0.015 | 0.009 | 0.0004 | 1000 |
| 127 | 0.03 | 0.036 | 0.016 | 0.0007 | 1000 |
| 175 | 0.05 | 0.046 | 0.023 | 0.001 | 1000 |
| 345 | 0.23 | 0.192 | 0.096 | 0.003 | 1000 |
| 436 | 0.382 | 0.292 | 0.124 | 0.005 | 1000 |
| 488 | 0.491 | 0.377 | 0.133 | 0.008 | 1000 |
| 493 | 0.573 | 0.438 | 0.137 | 0.01 | 1000 |
| 500 | 1.47 | 0.918 | 0.198 | 0.05 | 1000 |
| 500 | 2.637 | 1.5859 | 0.312 | 0.1 | 1000 |
| 500 | 4.37 | 2.582 | 0.496 | 0.18 | 1000 |
| 500 | 5.88 | 3.455 | 0.666 | 0.25 | 1000 |
| 500 | 8.57 | 4.937 | 0.860 | 0.36 | 1000 |
| 500 | 9.943 | 5.5669 | 1.039 | 0.42 | 1000 |
| 500 | 11.25 | 6.31 | 1.133 | 0.48 | 1000 |
| 500 | 13.143 | 7.346 | 1.335 | 0.56 | 1000 |
| 500 | 14.628 | 8.389 | 1.428 | 0.62 | 1000 |
| 500 | 15.7674 | 8.889 | 1.6013 | 0.68 | 1000 |
| 500 | 16.797 | 9.498 | 1.6959 | 0.72 | 1000 |
| 500 | 18.235 | 10.189 | 1.872 | 0.78 | 1000 |
| 500 | 19.40 | 11.12 | 1.896 | 0.82 | 1000 |
| 500 | 20.23 | 11.41 | 1.927 | 0.88 | 1000 |
| 500 | 21.58 | 12.0001 | 1.914 | 0.92 | 1000 |
| 500 | 22.665 | 12.9099 | 2.33 | 0.98 | 1000 |
| 500 | 23.06 | 12.957 | 2.3 | 1 | 1000 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Max Flow Value** | **Dinic’s Time** | **Edmond-Karp Time** | **Ford-Fulkerson Time** | **Density** | **Num of nodes** |
| 52 | 0.013 | 0.012 | 0.01 | 0.0001 | 1500 |
| 98 | 0.021 | 0.019 | 0.012 | 0.0002 | 1500 |
| 174 | 0.0559 | 0.047 | 0.025 | 0.0004 | 1500 |
| 272 | 0.119 | 0.099 | 0.042 | 0.0007 | 1500 |
| 337 | 0.21 | 0.183 | 0.082 | 0.001 | 1500 |
| 630 | 0.740 | 0.58 | 0.264 | 0.003 | 1500 |
| 727 | 1.12 | 0.836 | 0.26 | 0.005 | 1500 |
| 749 | 1.384 | 1.075 | 0.26 | 0.008 | 1500 |
| 747 | 1.521 | 1.177 | 0.278 | 0.01 | 1500 |
| 750 | 4.409 | 2.89 | 0.495 | 0.05 | 1500 |
| 750 | 8.55 | 5.29 | 0.766 | 0.1 | 1500 |
| 750 | 14.577 | 8.78 | 1.12 | 0.18 | 1500 |
| 750 | 19.783 | 11.53 | 1.595 | 0.25 | 1500 |
| 750 | 28.469 | 16.1 | 2.096 | 0.36 | 1500 |
| 750 | 32.959 | 18.72 | 2.534 | 0.42 | 1500 |
| 750 | 37.141 | 21.337 | 2.79 | 0.48 | 1500 |
| 750 | 43.219 | 24.013 | 3.3 | 0.56 | 1500 |
| 750 | 48.125 | 26.61 | 3.4 | 0.62 | 1500 |
| 750 | 52.568 | 29.25 | 3.87 | 0.68 | 1500 |
| 750 | 56.61 | 31.88 | 4.16 | 0.72 | 1500 |
| 750 | 61.154 | 36.473 | 4.74 | 0.78 | 1500 |
| 750 | 63.224 | 35.035 | 4.73 | 0.82 | 1500 |
| 750 | 68.044 | 37.47 | 5.26 | 0.88 | 1500 |
| 750 | 72.318 | 40.274 | 5.177 | 0.92 | 1500 |
| 750 | 75.187 | 42.28 | 5.579 | 0.98 | 1500 |
| 750 | 77.63 | 43.16 | 5.686 | 1 | 1500 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Max Flow Value** | **Dinic’s Time** | **Edmond-Karp Time** | **Ford-Fulkerson Time** | **Density** | **Num of nodes** |
| 94 | 0.0219 | 0.019 | 0.015 | 0.0001 | 2000 |
| 169 | 0.05 | 0.043 | 0.023 | 0.0002 | 2000 |
| 296 | 0.146 | 0.122 | 0.241 | 0.0004 | 2000 |
| 432 | 0.33 | 0.265 | 0.13 | 0.0007 | 2000 |
| 553 | 0.53 | 0.445 | 0.21 | 0.001 | 2000 |
| 927 | 1.735 | 1.532 | 0.462 | 0.003 | 2000 |
| 997 | 2.257 | 2.158 | 0.464 | 0.005 | 2000 |
| 1000 | 2.855 | 2.716 | 0.489 | 0.008 | 2000 |
| 1000 | 3.167 | 3.011 | 0.496 | 0.01 | 2000 |
| 1000 | 10.226 | 7.328 | 0.898 | 0.05 | 2000 |
| 1000 | 18.99 | 13.198 | 1.344 | 0.1 | 2000 |
| 1000 | 33.64 | 19.835 | 2.239 | 0.18 | 2000 |
| 1000 | 46.021 | 27.059 | 2.773 | 0.25 | 2000 |
| 1000 | 65.973 | 37.9 | 3.902 | 0.36 | 2000 |
| 1000 | 76.44 | 43.64 | 4.732 | 0.42 | 2000 |
| 1000 | 87.29 | 49.48 | 5.0526 | 0.48 | 2000 |
| 1000 | 101.27 | 57.089 | 5.881 | 0.56 | 2000 |
| 1000 | 114.89 | 63.63 | 6.574 | 0.62 | 2000 |
| 1000 | 123.89 | 70.326 | 8.0984 | 0.68 | 2000 |
| 1000 | 130.67 | 73.502 | 7.534 | 0.72 | 2000 |
| 1000 | 140.48 | 79.199 | 8.375 | 0.78 | 2000 |
| 1000 | 151.012 | 83.302 | 9.629 | 0.82 | 2000 |
| 1000 | 158.78 | 88.587 | 9.857 | 0.88 | 2000 |
| 1000 | 165.949 | 92.695 | 10.715 | 0.92 | 2000 |
| 1000 | 177.91 | 100.705 | 10.188 | 0.98 | 2000 |
| 1000 | 181.33 | 101.592 | 11.650 | 1 | 2000 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Max Flow Value** | **Dinic’s Time** | **Edmond-Karp Time** | **Ford-Fulkerson Time** | **Density** | **Num of nodes** |
| 145 | 0.039 | 0.037 | 0.022 | 0.0001 | 2500 |
| 245 | 0.1043 | 0.087 | 0.045 | 0.0002 | 2500 |
| 433 | 0.296 | 0.246 | 0.103 | 0.0004 | 2500 |
| 625 | 0.681 | 0.552 | 0.239 | 0.0007 | 2500 |
| 779 | 1.118 | 1.296 | 0.429 | 0.001 | 2500 |
| 1206 | 3.205 | 3.484 | 0.64 | 0.003 | 2500 |
| 1247 | 3.906 | 4.46 | 0.738 | 0.005 | 2500 |
| 1250 | 4.89 | 5.259 | 0.749 | 0.008 | 2500 |
| 1250 | 5.528 | 5.729 | 0.78 | 0.01 | 2500 |
| 1250 | 19.76 | 14.680 | 1.443 | 0.05 | 2500 |
| 1250 | 37.709 | 23.827 | 2.157 | 0.1 | 2500 |
| 1250 | 66.827 | 40.005 | 3.33 | 0.18 | 2500 |
| 1250 | 91.636 | 53.14 | 4.619 | 0.25 | 2500 |
| 1250 | 128.67 | 74.12 | 6.648 | 0.36 | 2500 |
| 1250 | 151.761 | 85.309 | 6.96 | 0.42 | 2500 |
| 1250 | 170.652 | 96.974 | 8.44 | 0.48 | 2500 |
| 1250 | 197.947 | 111.74 | 10.59 | 0.56 | 2500 |
| 1250 | 220.126 | 123.86 | 12.403 | 0.62 | 2500 |
| 1250 | 242.868 | 134.392 | 12.526 | 0.68 | 2500 |
| 1250 | 151.2 | 164.6 | 12.46 | 0.72 | 2500 |
| 1250 | 334.059 | 170.974 | 15.867 | 0.78 | 2500 |
| 1250 | 337.392 | 198.749 | 19.077 | 0.82 | 2500 |
| 1250 | 368.6 | 207.36 | 23.441 | 0.88 | 2500 |
| 1250 | 352.549 | 192.608 | 15.34 | 0.92 | 2500 |
| 1250 | 445.26 | 219.079 | 18.21 | 0.98 | 2500 |
| 1250 | 357.043 | 228.211 | 21.949 | 1 | 2500 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Max Flow Value** | **Dinic’s Time** | **Edmond-Karp Time** | **Ford-Fulkerson Time** | **Density** | **Num of nodes** |
| 194 | 0.062 | 0.051 | 0.037 | 0.0001 | 3000 |
| 355 | 0.186 | 0.149 | 0.071 | 0.0002 | 3000 |
| 566 | 0.525 | 0.419 | 0.194 | 0.0004 | 3000 |
| 842 | 1.129 | 0.936 | 0.42 | 0.0007 | 3000 |
| 1019 | 1.770 | 1.834 | 0.685 | 0.001 | 3000 |
| 1483 | 4.456 | 5.688 | 0.877 | 0.003 | 3000 |
| 1496 | 5.972 | 7.194 | 0.948 | 0.005 | 3000 |
| 1500 | 11.727 | 9.53 | 1.077 | 0.008 | 3000 |
| 1500 | 17.365 | 18.52 | 1.808 | 0.01 | 3000 |
| 1500 | 44.423 | 37.535 | 3.361 | 0.05 | 3000 |
| 1500 | 105.694 | 56.887 | 4.936 | 0.1 | 3000 |
| 1500 | 163.81 | 107.36 | 8.311 | 0.18 | 3000 |
| 1500 | 204.35 | 123.462 | 10.264 | 0.25 | 3000 |
| 1500 | 293.277 | 166.832 | 15.941 | 0.36 | 3000 |
| 1500 | 334.329 | 193.038 | 17.019 | 0.42 | 3000 |
| 1500 | 395.815 | 216.671 | 20.823 | 0.48 | 3000 |
| 1500 | 466.471 | 250.55 | 22.109 | 0.56 | 3000 |
| 1500 | 486.75 | 275.91 | 26.975 | 0.62 | 3000 |
| 1500 | 550.304 | 304.25 | 31.970 | 0.68 | 3000 |
| 1500 | 566.79 | 342.077 | 27.2663 | 0.72 | 3000 |
| 1500 | 637.237 | 334.254 | 29.541 | 0.78 | 3000 |
| 1500 | 650.677 | 364.236 | 33.845 | 0.82 | 3000 |
| 1500 | 694.883 | 384.40 | 43.368 | 0.88 | 3000 |
| 1500 | 730.825 | 403.37 | 43.497 | 0.92 | 3000 |
| 1500 | 805.852 | 428.80 | 42.252 | 0.98 | 3000 |
| 1500 | 785.558 | 444.132 | 41.365 | 1 | 3000 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Max Flow Value** | **Dinic’s Time** | **Edmond-Karp Time** | **Ford-Fulkerson Time** | **Density** | **Num of nodes** |
| 266 | 0.473 | 0.345 | 0.196 | 0.0001 | 3500 |
| 459 | 1.934 | 1.402 | 0.656 | 0.0002 | 3500 |
| 771 | 2.331 | 2.107 | 1.072 | 0.0004 | 3500 |
| 1084 | 5.654 | 11.347 | 3.05 | 0.0007 | 3500 |
| 1297 | 4.240 | 5.416 | 1.692 | 0.001 | 3500 |
| 1742 | 7.86 | 13.55 | 2.164 | 0.003 | 3500 |
| 1750 | 9.987 | 13.975 | 1.582 | 0.005 | 3500 |
| 1750 | 12.541 | 16.357 | 1.749 | 0.008 | 3500 |
| 1750 | 16.375 | 16.163 | 1.7143 | 0.01 | 3500 |
| 1750 | 84.475 | 85.325 | 4.181 | 0.05 | 3500 |
| 1750 | 149.177 | 99.195 | 7.32 | 0.1 | 3500 |
| 1750 | 229.326 | 152.34 | 11.895 | 0.18 | 3500 |
| 1750 | 351.389 | 204.047 | 16.493 | 0.25 | 3500 |
| 1750 | 438.027 | 270.007 | 23.274 | 0.36 | 3500 |
| 1750 | 401.626 | 287.403 | 17.11 | 0.42 | 3500 |
| 1750 | 455.124 | 271.93 | 18.14 | 0.48 | 3500 |
| 1750 | 559.934 | 310.39 | 23.419 | 0.56 | 3500 |
| 1750 | 582.746 | 336.792 | 25.001 | 0.62 | 3500 |
| 1750 | 640.794 | 373.088 | 27.173 | 0.68 | 3500 |
| 1750 | 674.528 | 389.051 | 33.315 | 0.72 | 3500 |
| 1750 | 734.0134 | 419.169 | 31.645 | 0.78 | 3500 |
| 1750 | 773.341 | 443.416 | 33.350 | 0.82 | 3500 |
| 1750 | 833.635 | 473.7629 | 38.372 | 0.88 | 3500 |
| 1750 | 871.416 | 508.414 | 38.327 | 0.92 | 3500 |
| 1750 | 928.698 | 529.56 | 42.978 | 0.98 | 3500 |
| 1750 | 953.052 | 544.425 | 42.405 | 1 | 3500 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Max Flow Value** | **Dinic’s Time** | **Edmond-Karp Time** | **Ford-Fulkerson Time** | **Density** | **Num of nodes** |
| 342 | 0.18 | 0.161 | 0.082 | 0.0001 | 4000 |
| 583 | 0.563 | 0.485 | 0.224 | 0.0002 | 4000 |
| 958 | 1.678 | 1.634 | 0.657 | 0.0004 | 4000 |
| 1325 | 3.268 | 4.314 | 1.317 | 0.0007 | 4000 |
| 1569 | 4.926 | 7.281 | 1.831 | 0.001 | 4000 |
| 1991 | 10.952 | 15.868 | 1.95 | 0.003 | 4000 |
| 2000 | 13.303 | 18.963 | 3.42 | 0.005 | 4000 |
| 2000 | 17.658 | 21.854 | 2.103 | 0.008 | 4000 |
| 2000 | 20.747 | 23.223 | 2.27 | 0.01 | 4000 |
| 2000 | 78.654 | 59.927 | 3.986 | 0.05 | 4000 |
| 2000 | 151.019 | 99.455 | 6.552 | 0.1 | 4000 |
| 2000 | 264.846 | 163.411 | 11.054 | 0.18 | 4000 |
| 2000 | 362.043 | 219.860 | 15.44 | 0.25 | 4000 |
| 2000 | 517.476 | 305.978 | 19.704 | 0.36 | 4000 |
| 2000 | 663.042 | 353.449 | 25.395 | 0.42 | 4000 |
| 2000 | 687.149 | 395.681 | 30.491 | 0.48 | 4000 |
| 2000 | 799.901 | 461.467 | 37.541 | 0.56 | 4000 |
| 2000 | 1130.312 | 694.595 | 61.039 | 0.62 | 4000 |
| 2000 | 968.718 | 585.908 | 45.451 | 0.68 | 4000 |
| 2000 | 1010.334 | 583.5712 | 47.494 | 0.72 | 4000 |
| 2000 | 1047.558 | 581.4655 | 40.28 | 0.78 | 4000 |
| 2000 | 1099.4127 | 625.728 | 39.184 | 0.82 | 4000 |
| 2000 | 1194.743 | 656.953 | 49.0127 | 0.88 | 4000 |
| 2000 | 1244.624 | 698.317 | 49.889 | 0.92 | 4000 |
| 2000 | 1339.7557 | 738.556 | 50.986 | 0.98 | 4000 |
| 2000 | 1346.0877 | 759.0217 | 54.039 | 1 | 4000 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **BA** | | **LMM** | | **DMM** | | **SMD** | | **RR** | | **SG** | | **MC** | | **G** | | |
| **T** | **R** | **T** | **R** | **T** | **R** | **T** | **R** | **T** | **R** | **T** | **R** | **T** | **R** | **MM** | **D** | **N** |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0.0001 | 6000 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0.0002 | 6000 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0.0004 | 6000 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0.0007 | 6000 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0.001 | 6000 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0.003 | 6000 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0.005 | 6000 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0.008 | 6000 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0.01 | 6000 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0.05 | 6000 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0.1 | 6000 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0.18 | 6000 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0.25 | 6000 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0.36 | 6000 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0.42 | 6000 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0.48 | 6000 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0.56 | 6000 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0.62 | 6000 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0.68 | 6000 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0.72 | 6000 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0.78 | 6000 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0.82 | 6000 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0.88 | 6000 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0.92 | 6000 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0.98 | 6000 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 | 6000 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |