



2023

知识图谱的查询优化新技术

李友焕
湖南大学

liyuhuan@hnu.edu.cn



目录

CONTENTS

01 从图数据模型
PART 到知识图谱

03 实体间关联查
PART 询优化新技术

02 基础图模式 (BGP)
PART 的查询优化新技术

04 总结
PART SUMMARY

01
PART

从图数据模型
到知识图谱



传统的关系数据模型

找出X的所有好友

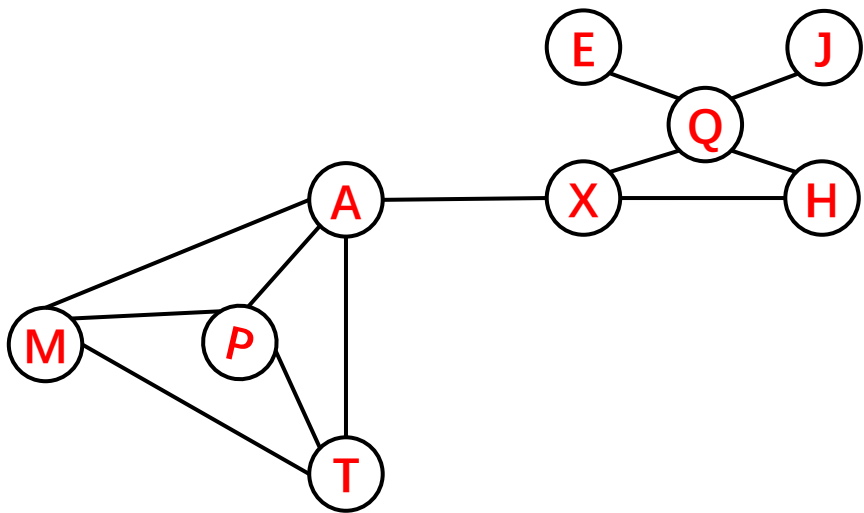
找出朋友不止一个的人

找出所有的三人组，
其中三个人互相认识

传统好友关系表

Name1	Name2
A	M
A	P
A	T
A	X
E	Q
H	X
H	Q
J	Q
M	P
M	T
Q	X
P	T

图谱数据模型



邻接表

A → M、P、T、X
E → Q
H → X、Q
J → Q
M → P、T
Q → E、H、J、X
P → A、T、M
T → A、M、P
X → A、Q、H

找出朋友不止一个的人

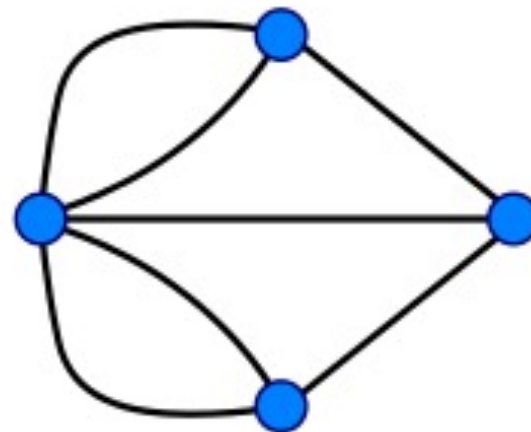
找出所有的三人组，其中三个人互相认识

图论起源：欧拉七桥问题

欧拉七桥问题：如何能够不走重复路的情况下走遍哥尼斯堡的七座桥？

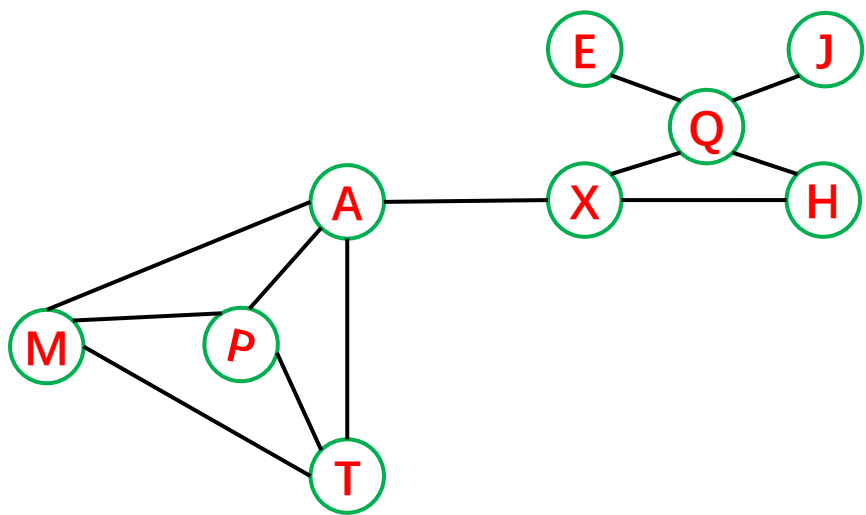
图论思路：欧拉通过将七桥问题形式化为点边的一笔画问题来解决。

Leonhard Euler



图形式简单，图技术复杂

点、边、度数



绿色是点，对应现实世界对象

点间连的黑线是边，对应现实世界对象之间的关联

度数是定义在每个点上的一个数值，即跟该点有连边的其它点的数量

请求出所有顶点的度数

一笔画体验图谱拓扑、结论

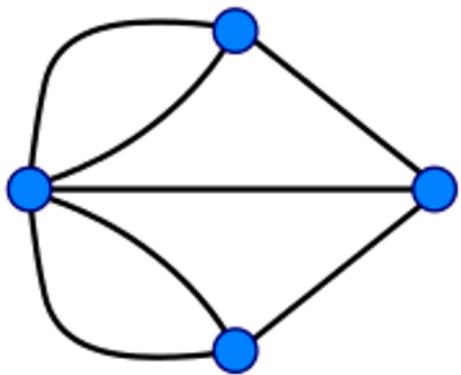


图 A

1. 图形必须是**连通**的。
2. 图中的“奇度数点”个数是0或2。

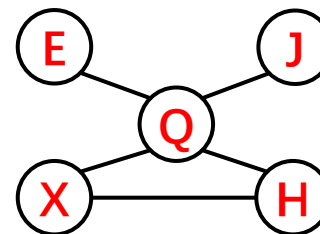
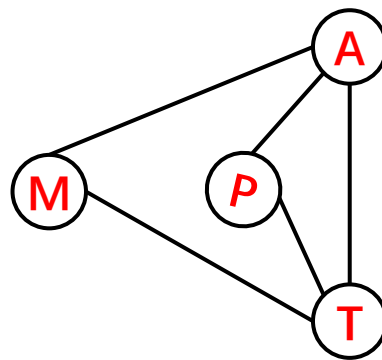


图 B

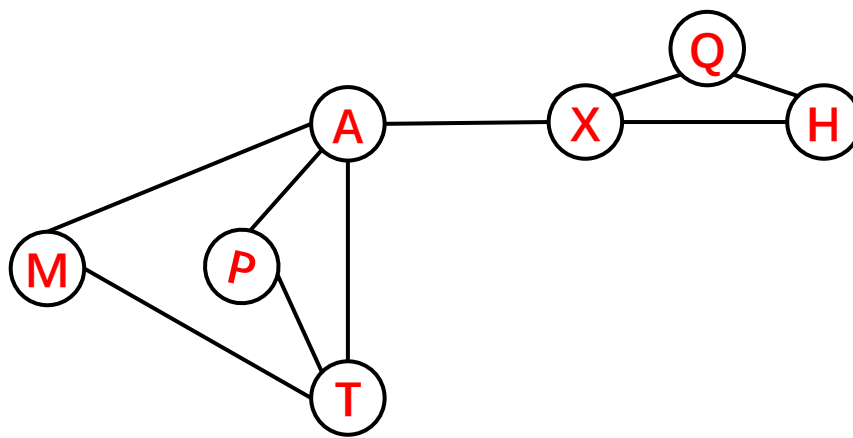


图 C

图的形式化定义

图论中，图被定义为一个多元组

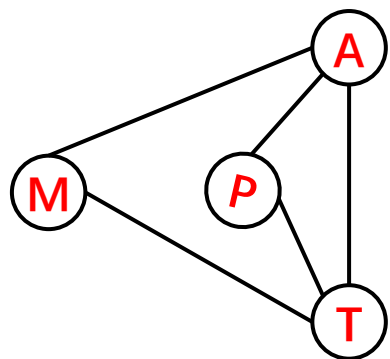
$$G = (V, E, L, P \dots)$$

V 为点 (vertex) 集,
对应对象的集合

$E \subseteq V \times V$ 为边
(edge) 集, 对应
对象间关系、交互
的集合

L 为点/边标签函数
(可选)
 $L(v1) = \text{'诈骗'}$,
 $L(e1) = \text{'好友'}$

P 为点/边属性函数
(可选)
 $P(v1) = 28$,
 $P(e1) = 20201125$



$$G = (V, E), \quad V = \{A, P, T, M\}, \quad E = \{(A, P), (A, T), (A, M), (T, P), (T, M)\}$$

点、边均只有一种类型的，为同构图，否则为异构图

知识的重要表达方式：陈述句

屠呦呦，女，1930年12月30日出生于浙江宁波 [78]，汉族，中共党员，药学家。1951年考入北京大学医学院药学系生药专业 [1-2]。1955年毕业于北京医学院（今北京大学医学部），毕业后接受中医培训两年半，并一直在中国中医研究院（2005年更名为中国中医科学院）工作，期间晋升为硕士生导师、博士生导师。现为中国中医科学院首席科学家 [3-5]，终身研究员兼首席研究员 [6]，青蒿素研究开发中心主任，博士生导师，共和国勋章获得者。 [7]



- 屠呦呦的性别是女
- 屠呦呦出生于浙江宁波
- 屠呦呦出生日期是1930年12月30日
- 屠呦呦的民族是汉族
- 屠呦呦的党派是中国共产党
- 屠呦呦的学术身份是药学家
- 屠呦呦毕业于北京大学医学部
- 屠呦呦就职于中国中医研究院
- 屠呦呦的荣誉有共和国勋章获得者

主语	谓语	宾语
屠呦呦	性别	女
屠呦呦	出生地	浙江宁波
屠呦呦	出生日期	1930年12月30日
屠呦呦	民族	汉族
屠呦呦	党派	中国共产党
屠呦呦	学术身份	药学家
屠呦呦	毕业院校	北京大学医学部
屠呦呦	单位	中国中医研究院
屠呦呦	荣誉含有	共和国勋章获得者

问句 → 主谓 ? 宾

屠呦呦获得哪些奖项？

屠呦呦 获奖 ?

诺贝尔生理学或医学奖

国家最高科学技术奖

改革先锋奖章

三元组：三个基本元素

屠呦呦毕业于北京大学医学部

陈述句的三元组

主谓宾：

屠呦呦 毕业院校 北京大学医学部

疑问句的三元组

①主谓？：

(屠呦呦毕业于哪里?) 屠呦呦 毕业院校 ?

②主 ? 宾：

(屠呦呦跟北大医学部什么关系?) 屠呦呦 ? 北京大学医学部

③? 谓 宾：

(哪些人毕业于北京大学医学部?) ? 毕业院校 北京大学医学部

④主 ? ? ？：

(屠呦呦的信息有哪些?) 屠呦呦 ??

... ..

图谱元素建模信息中对象及其间的关联

主语	谓语	宾语
屠呦呦	性别	女
屠呦呦	出生地	浙江宁波
屠呦呦	出生日期	1930年12月30日
屠呦呦	民族	汉族
屠呦呦	党派	中国共产党
屠呦呦	学术身份	药学家
屠呦呦	毕业院校	北京大学医学部
屠呦呦	单位	中国中医研究院
屠呦呦	荣誉含有	共和国勋章

主语、宾语为对象
谓语表达对象间的关联

$$G = (V, E)$$

V
= {屠呦呦, 女, 浙江宁波, 1930年12月30日, 汉族, 中国共产党, 药学家, 北京大学医学部, 中国中医研究院, 共和国勋章}

E = {(屠呦呦, 女), (屠呦呦, 浙江宁波),}

图谱模型对知识的表达、融合和分析

信息

图

图模型 $G = (V, E)$

- 对象 $\rightarrow$ 点
- 对象间关联 $\rightarrow$ 边



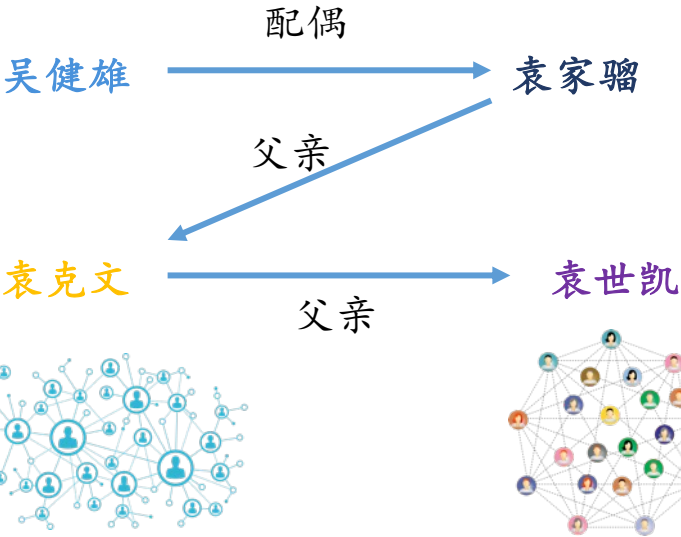
信息表达

- 知识/信息 $\rightarrow$ 陈述句
- 陈述句 $\rightarrow$ 主谓宾

.....
吴健雄配偶袁家骝
袁家骝的父亲是袁克文
袁克文的父亲是袁世凯
.....

关联分析

- 对象间关联
- 多对象复杂关联



复杂知识/信息的建模、表达与融合

高效的关联发现与分析

搜索引擎

搜索引擎工作示意：

第一步：提出问题 老家在安徽且在国务院工作的一个博士？

第二步：引入搜索引擎



第三步：识别关键字



第四步：返回搜索结果



第五步：页面结果人工介入

李克强 (1955年7月1日 -)，男，汉族，安徽定远人，祖籍江西武宁^[3]，生于安徽合肥，是现任中央政治局常务委员会委员 由中华人民共和国国务院

1 关键字表达简单

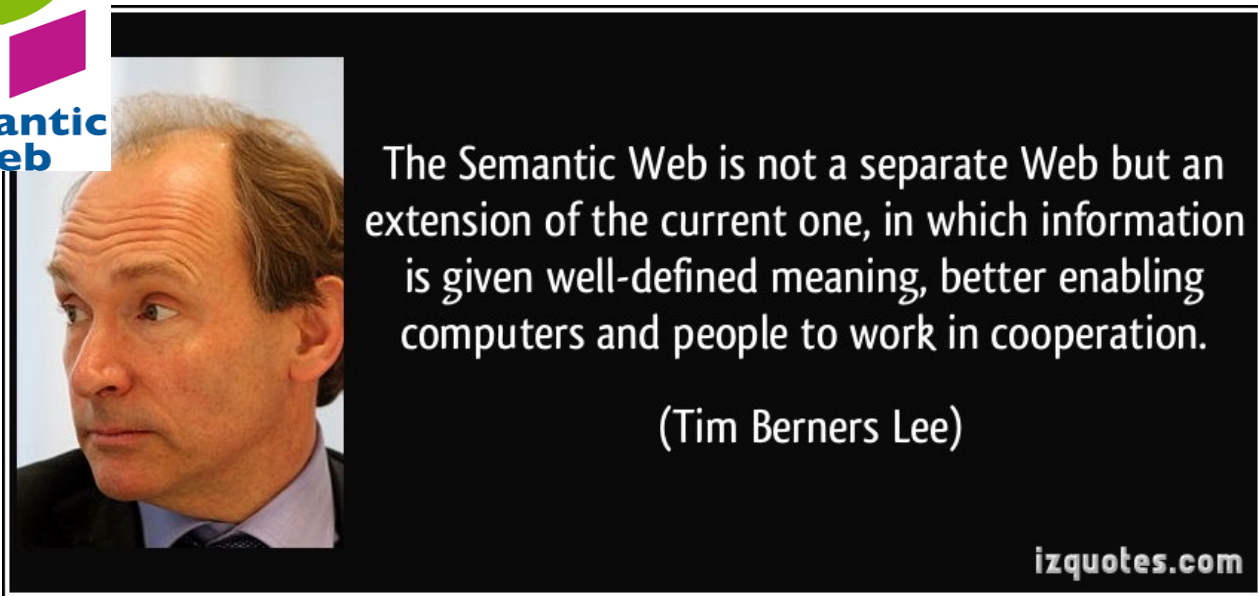
- 易出现关键词歧义，如在安徽工作

2 返回结果不直接

- 出现多个返回页面，需人工介入处理

语义网：如何解决搜索引擎遇到的问题

语义网：旨在将文档上的元素添加计算机能理解的语义，使得互联网成为一个通用的信息交换介质。



语义网重要标准：RDF和SPARQL



信息的三元组表达

图模型 $G = (V, E)$

- 对象 $\rightarrow$ 点
- 对象间关联 $\rightarrow$ 边



信息表达

- 知识/信息 $\rightarrow$ 陈述句
- 陈述句 $\rightarrow$ 主谓宾

.....

吴健雄配偶袁家骝

袁家骝的父亲是袁克文

袁克文的父亲是袁世凯

.....

对知识的表达不限领域、主题，融合能力强

RDF: Resource Description Framework

李克强 (1955年7月1日 -)，男，汉族，安徽定远人，祖籍江西武宁^[3]，生于安徽合肥，是现任中央政治局常务委员会委员、中华人民共和国国务院总

<李克强> 毕业于 <北京大学> .
<李克强> 学位 "经济学博士" .
<李克强> 家乡 <安徽省> .
<李克强> 单位 <中华人民共和国国务院> .
<李克强> 出生日期 "1955-07-01" .
<李克强> 性别 "男" .
<李克强> 民族 "汉" .

语义网标准：SPARQL

老家在安徽且在国务院工作的一个博士?



SPARQL (RDF Query Language)

```
1 select ?p from knowledgeGraph1
2 where {
3     ?p 学历 "博士" .
4     ?p 单位 <中共中央国务院> .
5     ?p 籍贯 <安徽省> .
6 }
```

知识图谱查询三阶段

老家在安徽并且在国
务院工作的一个博士 ?

① 将问句翻译成 SPARQL

SPARQL (RDF Query Language)

```
1 select ?p from knowledgeGraph1
2 where {
3     ?p 学历 "博士" .
4     ?p 单位 <中共中央国务院> .
5     ?p 籍贯 <安徽省> .
6 }
```



③ 处理 SPARQL 到 RDF
的 查询

查询

李克强 (1955年7月1日 -)，男，汉族，安徽定远人，祖籍江西武宁^[3]，生于安徽合肥，是现任中央政治局常务委员会委员、中华人民共和国国务院总理及现任主要领导人之一，毕业于北京大学法律系，拥有法学学士、经济学博士学位，在职研究生

② 将数据建模成 RDF

```
<李克强> 毕业于 <北京大学> .
<李克强> 学位 "经济学博士" .
<李克强> 家乡 <安徽省> .
<李克强> 单位 <中华人民共和国国务院> .
<李克强> 出生日期 "1955-07-01" .
<李克强> 性别 "男" .
<李克强> 民族 "汉" .
```

RDF (Resource Description Framework)

语义网到知识图谱

RDF、SPARQL的核心：三元组，对象与对象间的关联。

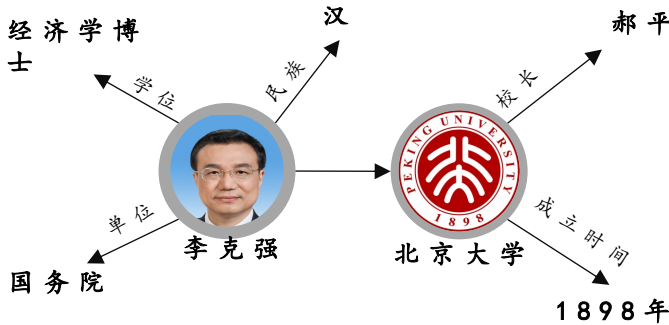
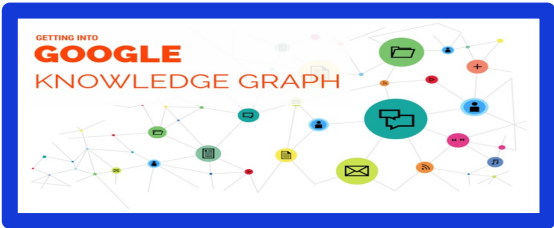
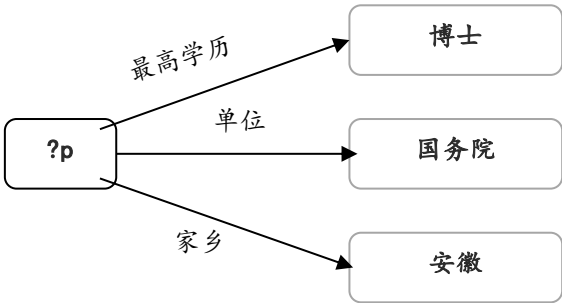
SPARQL (RDF Query Language)

```
1 select ?p from knowledgeGraph1
2 where {
3     ?p 学历 "博士" .
4     ?p 单位 <中共中央国务院> .
5     ?p 籍贯 <安徽省> .
6 }
```



<李克强> 毕业于 <北京大学> .
<李克强> 学位 "经济学博士" .
<李克强> 家乡 <安徽省> .
<李克强> 单位 <中华人民共和国国务院> .
<李克强> 出生日期 "1955-07-01" .
<李克强> 性别 "男" .
<李克强> 民族 "汉" .

RDF (Resource Description Framework)



知识图谱

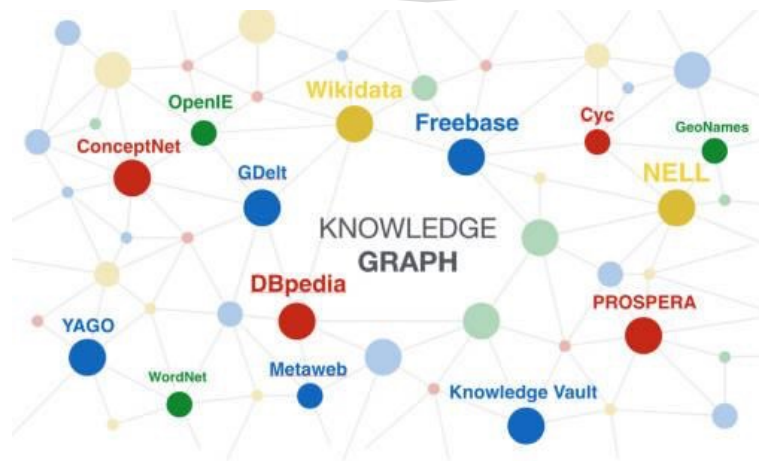
融合知识和图谱的管理、分析方法，进而对实际应用输出解决方案的一整套技术体系。

图谱

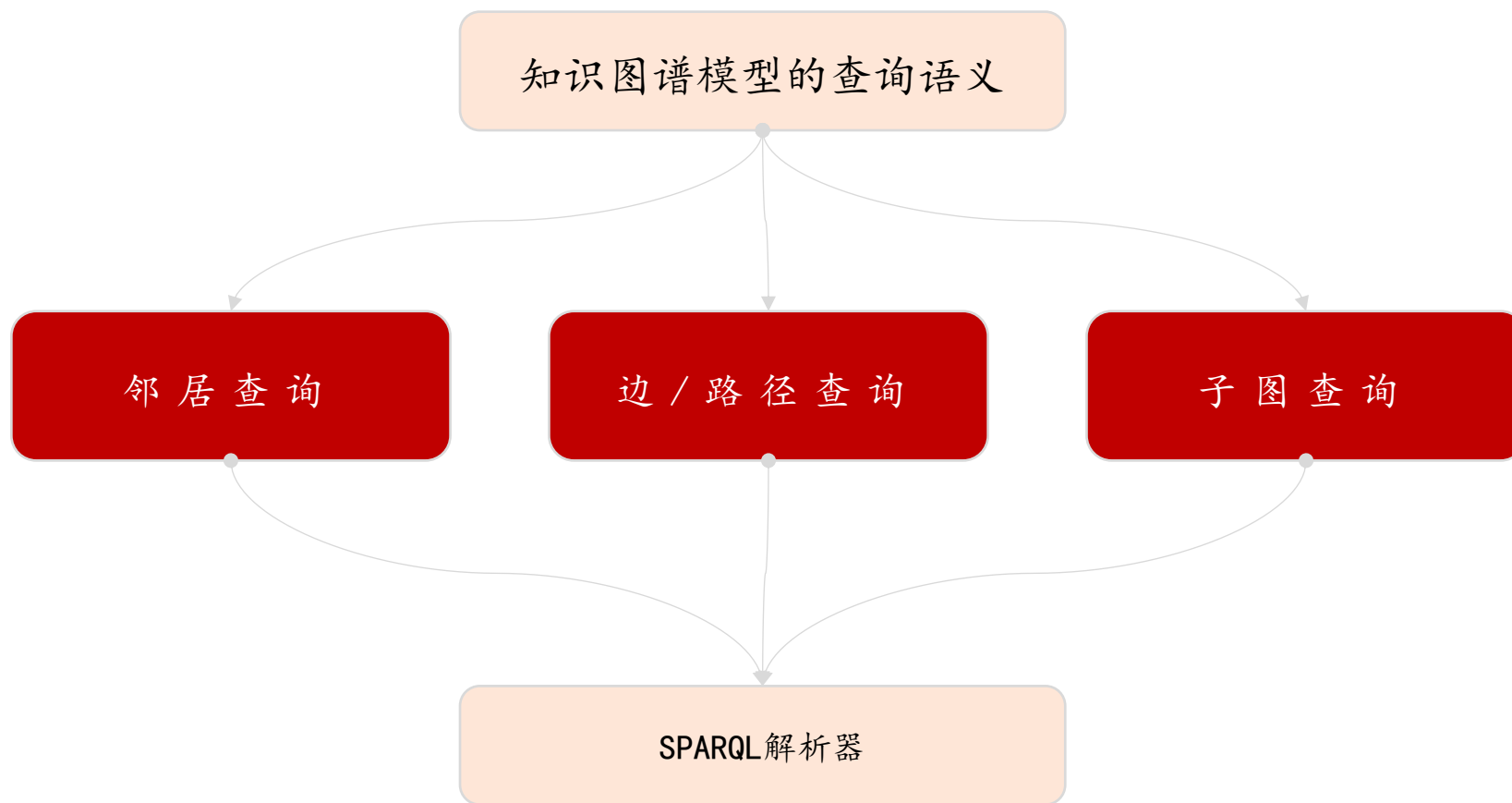


+

知识

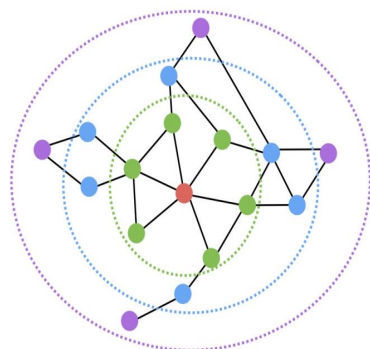


知识图谱的查询语义



多阶邻居查询

多阶邻居查询示例：

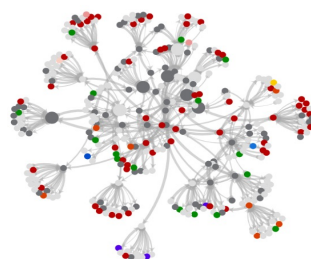


- **Red:** Target node
- **Green:** 1-hop neighbors
 - A (i.e., adjacency matrix)
- **Blue:** 2-hop neighbors
 - A^2
- **Purple:** 3-hop neighbors
 - A^3

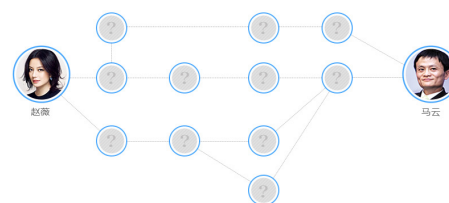
团伙挖掘



近邻分布



关联全像



二阶人脉



SPARQL 查询：

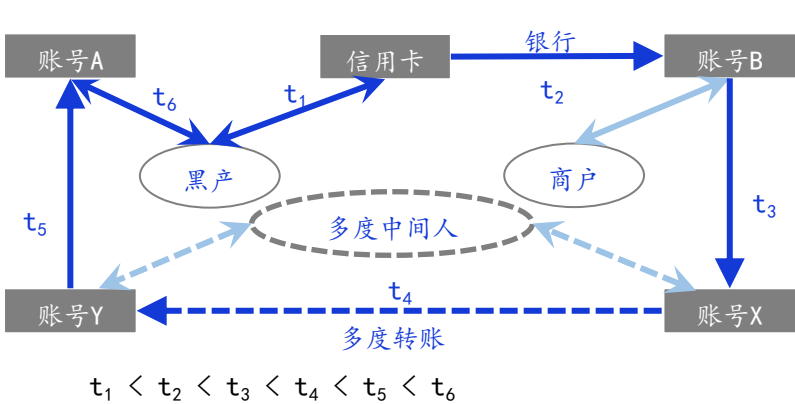
```
select ?onehop from KnowledgeGraph1 where {  
  <马化腾> ?relation1 ?onehop .  
}
```

```
select ?twohop from KnowledgeGraph1 where {  
  <马化腾> ?relation1 ?onehop .  
  ?onehop ?relation2 ?twohop .  
}
```

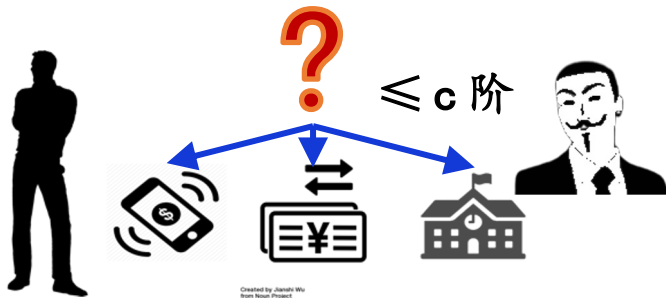
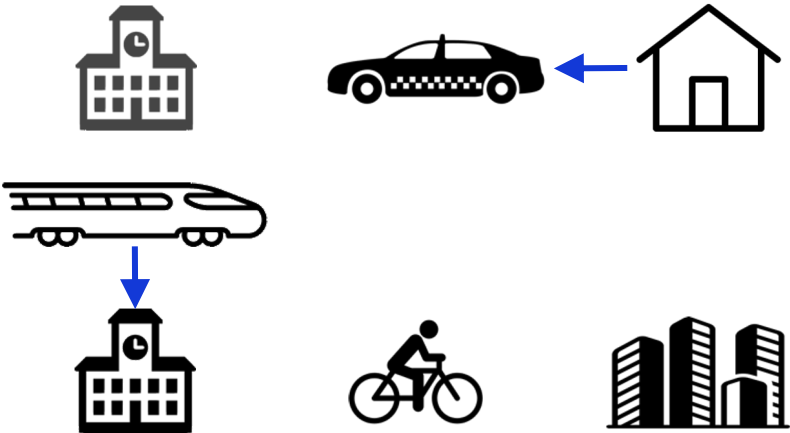
路径查询

路径查询示例：

反欺诈-套现



交通规划轨迹分析



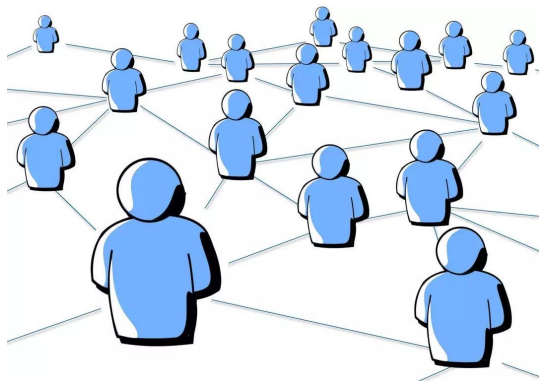
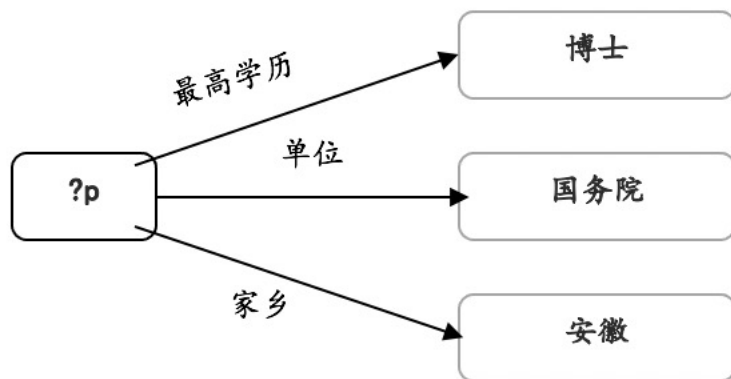
SPARQL 查询：

```
select ?twohop from KnowledgeGraph1 where {  
  <吴健雄> ?relation1 ?onehop .  
  ?onehop ?relation2 ?twohop .  
  ?twohop ?relation3 <袁世凯> .  
}
```

也支持不定长表达式

子图查询

子图查询示例：



gStoreParser

SPARQL 查询：

```
1 select ?p from knowledgeGraph1
2 where {
3     ?p 学历 "博士" .
4     ?p 单位 <中共中央国务院> .
5     ?p 籍贯 <安徽省> .
6 }
```

基础图模式 (BGP)：

知识图谱查询研究中最主要的优化对象

02 PART

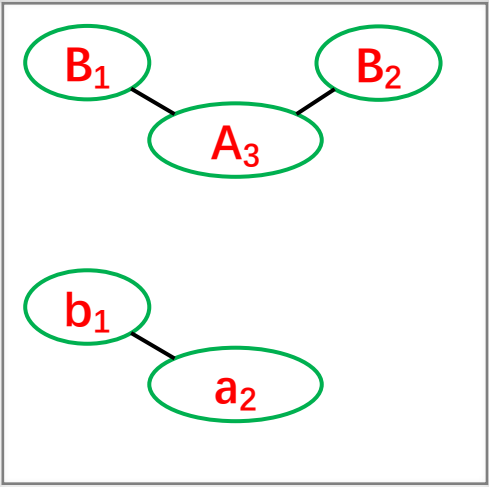
基础图模式（BGP）的 查询优化新技术



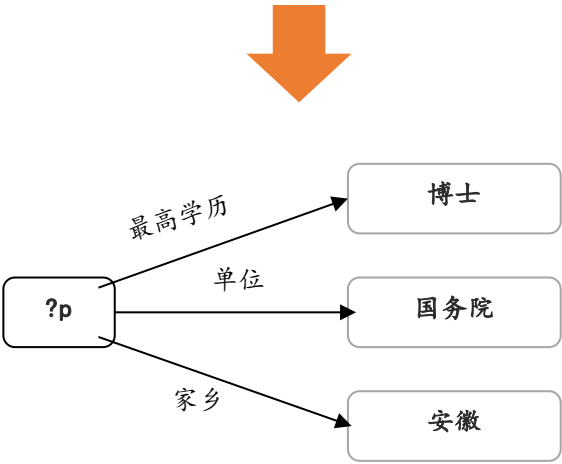
BGP查询本质是子图同态

子图同态

- 给定查询图Q和一个数据图G的子图g,
- 如果存在映射F从 $V(Q)$ 到 $V(g)$, 满足:
 - 任何 $u \in V(Q)$, u匹配F(u) (如标签相同)
 - 任何 $(u, v) \in E(Q)$, 则有 $(F(u), F(v)) \in E(g)$
- 则成g为Q在G上的一个同态子图
- 子图同构是一种特殊的子图同态
 - F是双射
 - $(u, v) \in E(Q)$ 当且仅当 $(F(u), F(v)) \in E(g)$



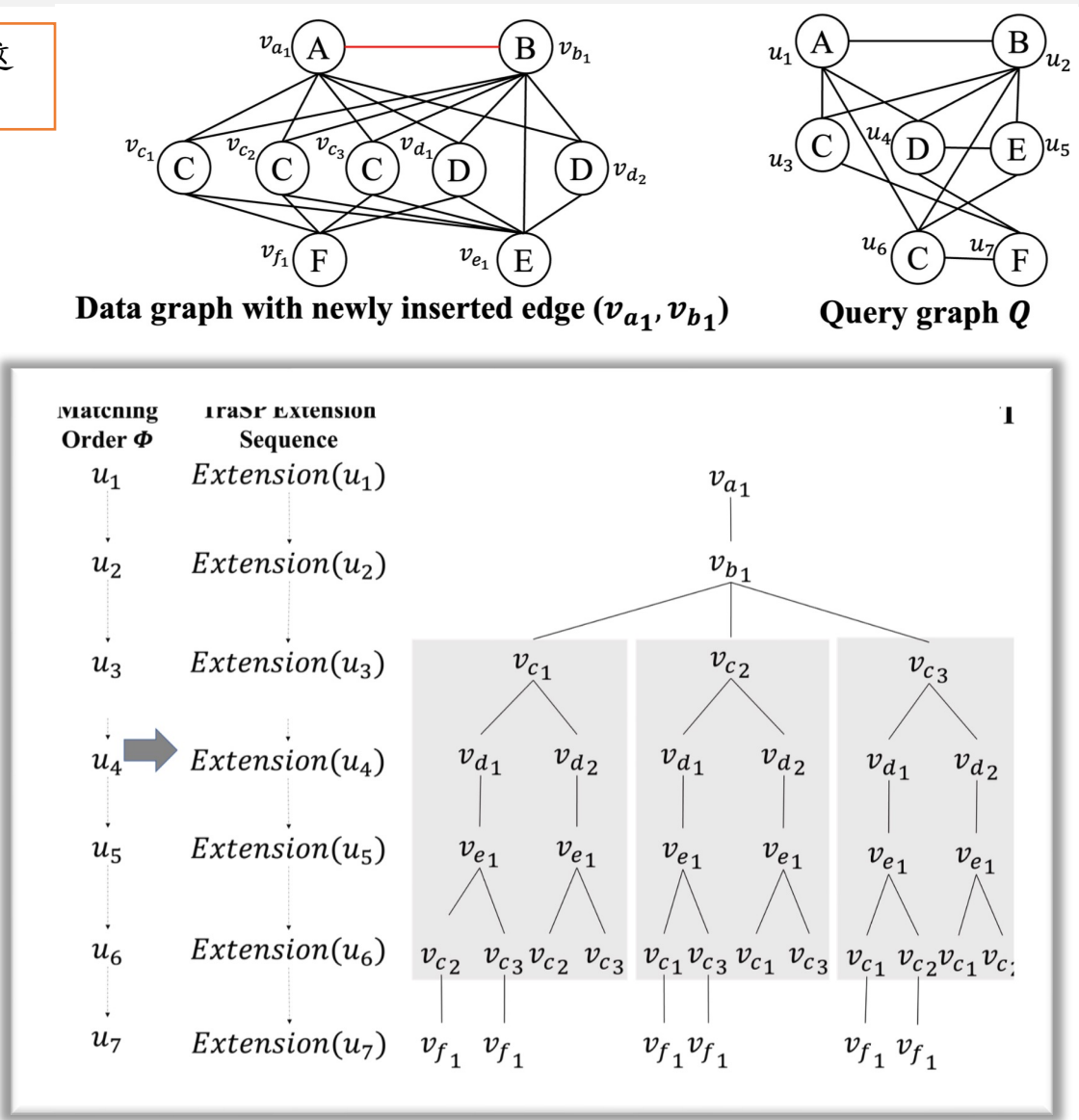
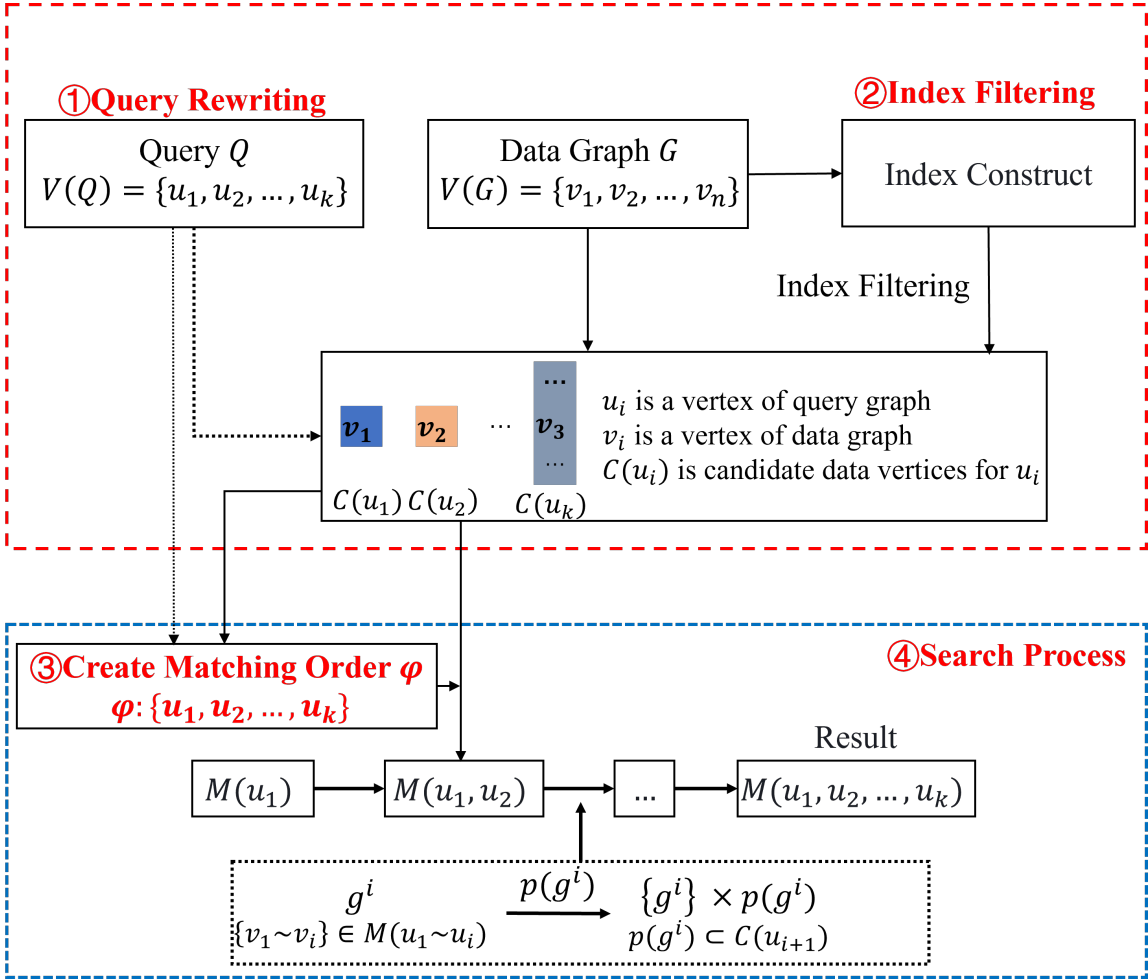
```
1 select ?p from knowledgeGraph1
2 where {
3     ?p 学历 "博士" .
4     ?p 单位 <中共中央国务院> .
5     ?p 籍贯 <安徽省> .
6 }
```



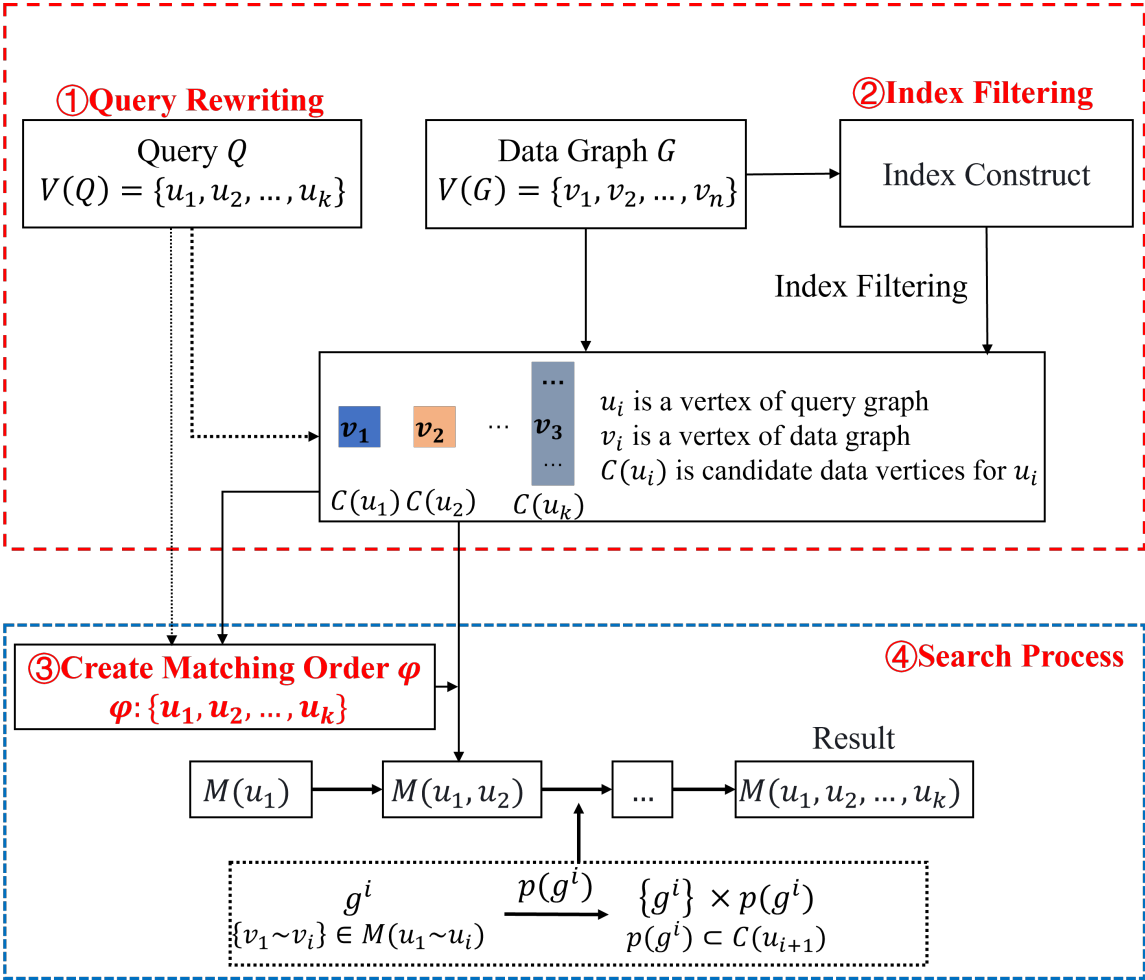
- 子图同构和子图同态是两种最为保持结构关系的子图匹配，匹配约束上，子图同构强于子图同态
- 两者在现有查询技术体系上高度相似，在子图同态结果上增加一层双射过滤即可快速获得子图同构。

BGP 查询的经典技术框架

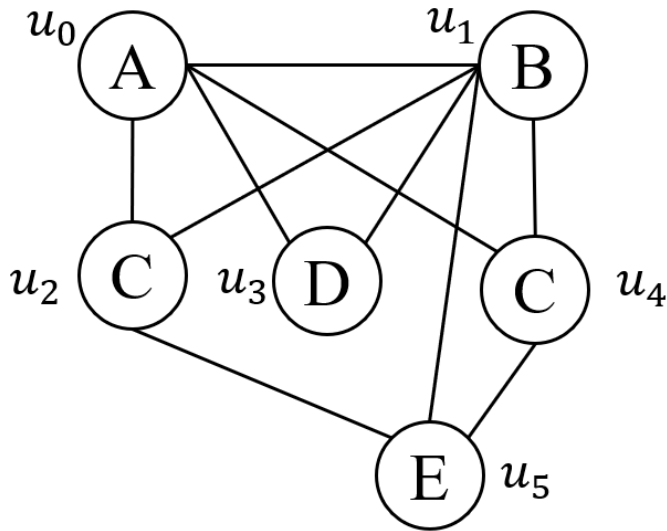
➤ 传统知识图谱BGP查询技术主要集中在查询重写、索引设计和匹配序生成，在这三个方面的技术确定之后，后续搜索过程中的剪枝和加速策略也相对直接。



BGP 查询的查询重写



➤ ① 查询重写
结合查询图结构进行图转化



TurboISO
Han et al. 2013

BGP 查询的索引过滤

① Query Rewriting

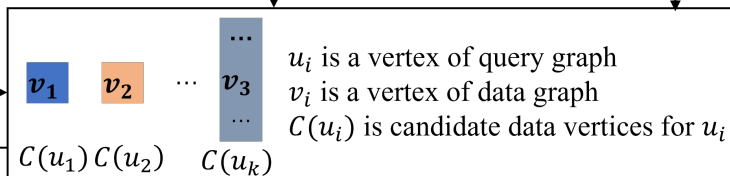
Query Q
 $V(Q) = \{u_1, u_2, \dots, u_k\}$

② Index Filtering

Data Graph G
 $V(G) = \{v_1, v_2, \dots, v_n\}$

Index Construct

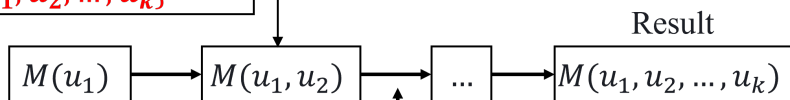
Index Filtering



③ Create Matching Order φ

$\varphi: \{u_1, u_2, \dots, u_k\}$

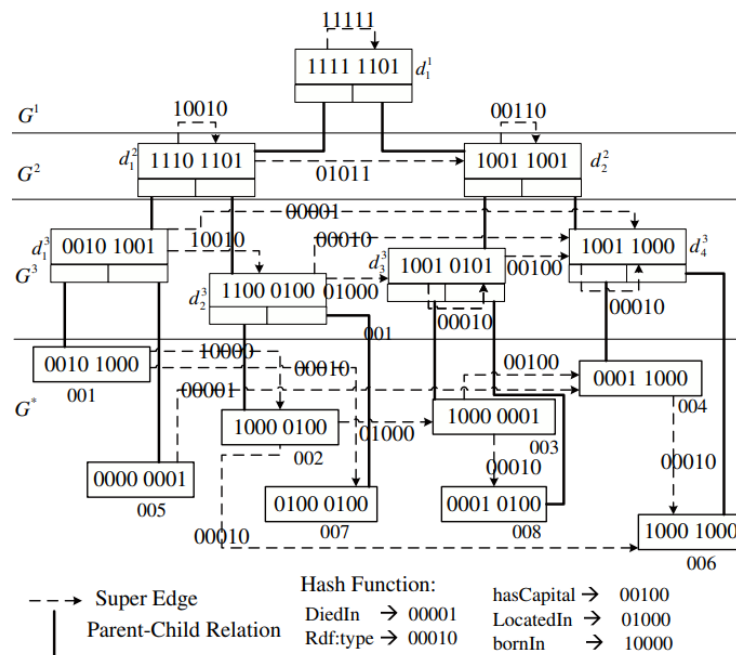
④ Search Process



g^i
 $\{v_1 \sim v_i\} \in M(u_1 \sim u_i)$
 $p(g^i) \rightarrow \{g^i\} \times p(g^i)$
 $p(g^i) \subset C(u_{i+1})$

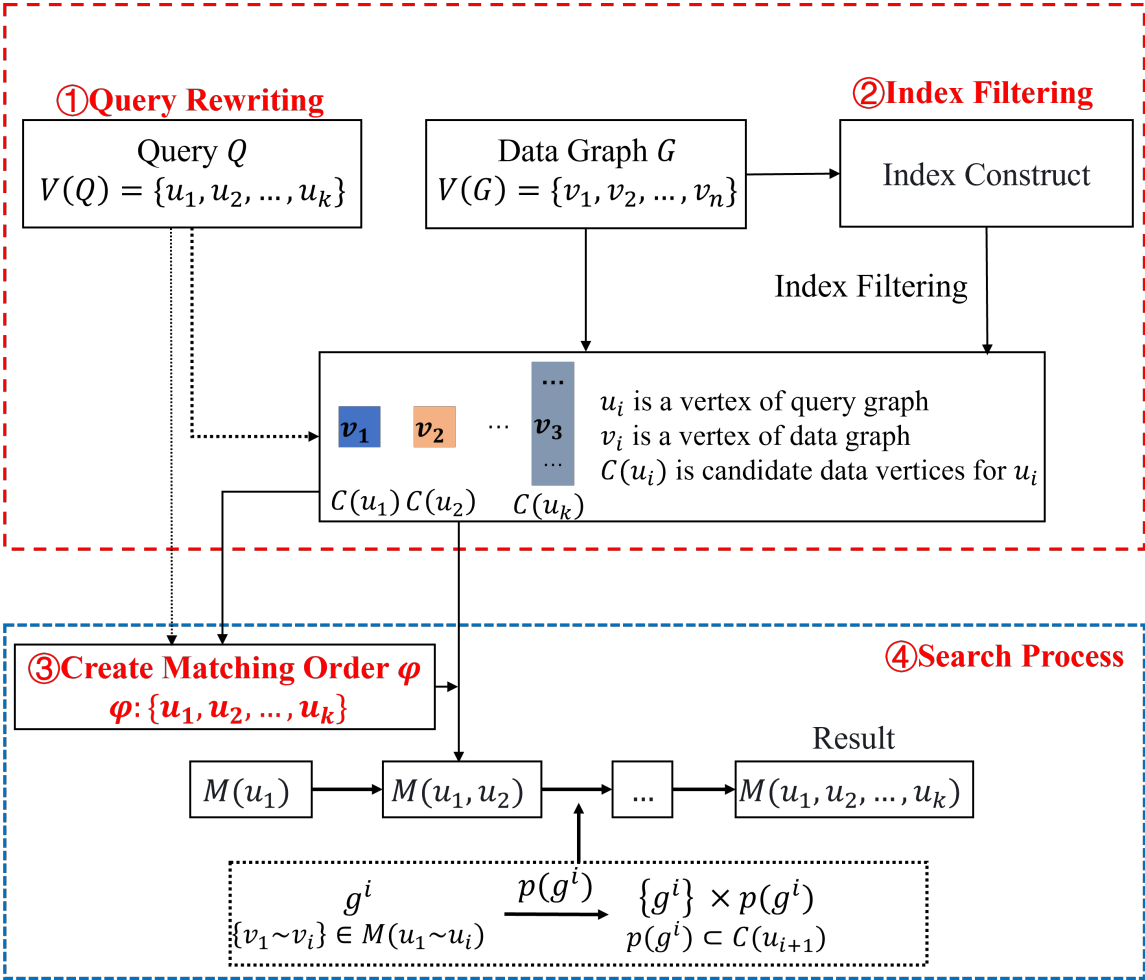
➤ ② 索引过滤

- 基于数据图构建复杂的离线索引
- 结合查询图以及索引进行最大可能的过滤，得到候选点、边
- 策略有：邻居标签过滤（NLF），节点邻居树（Node Neighbor Tree），近邻结构哈希编码（如gStore）等

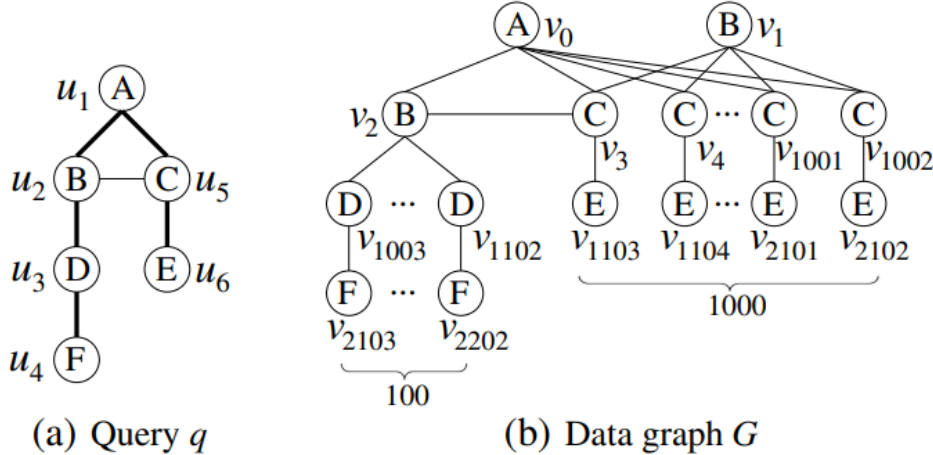


Zou et al. 2011

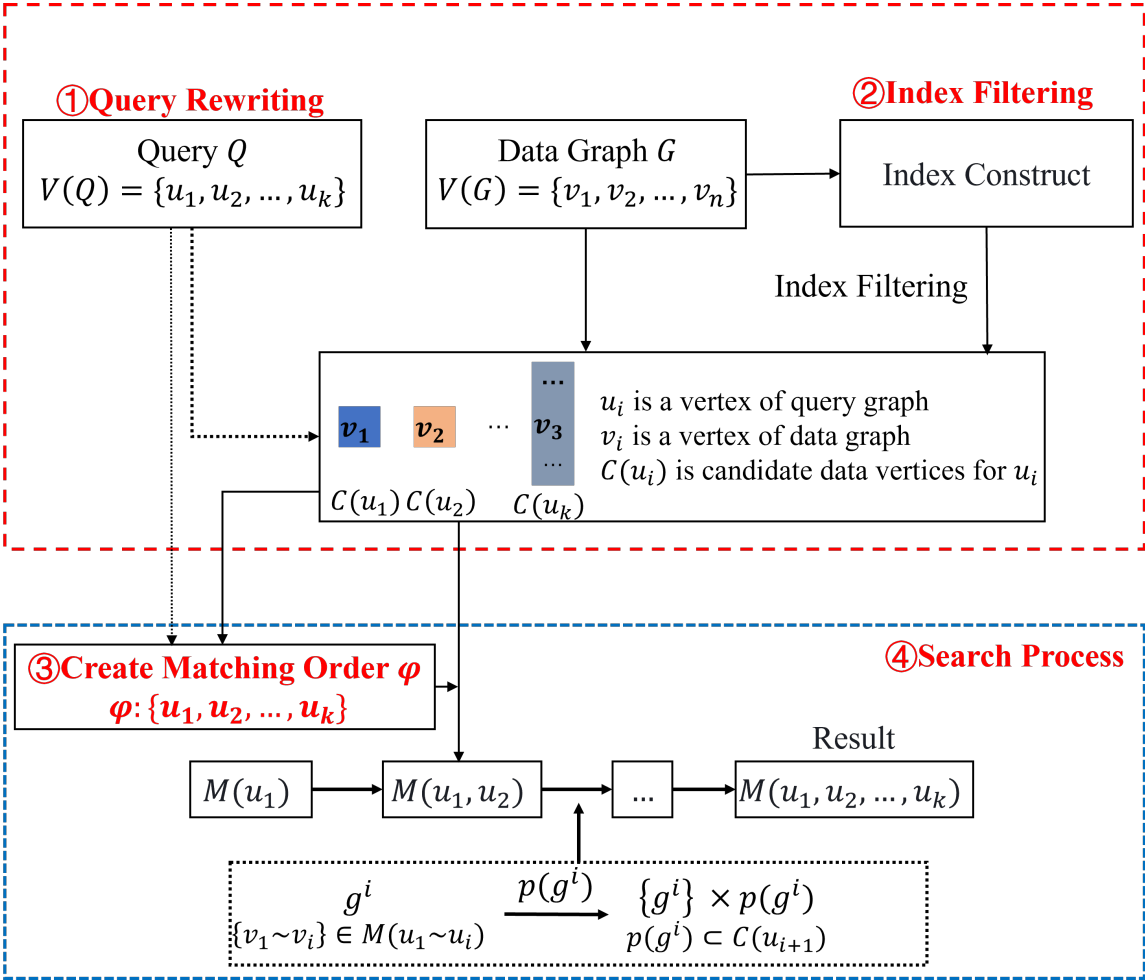
BGP 查询的匹配序生成



- ③匹配序生成 (Matching Order Generation)
 - 制定将部分解不断扩大成为最终解的拼接顺序, 最小的部分解是单个查询点和一个数据点形成的匹配关系
 - 每次选择一个查询点 (及其候选点集) 对现有部分解进行扩展 (size增1)
 - 最终解规模固定, 但中间解规模的大小 (即搜索空间大小) 决定计算代价, 越小的中间解表示计算越优
 - 策略有: 优先扩展小的候选解、推迟单度数点以避免笛卡尔积、基于代价模型的动态匹配序等



BGP 查询的搜索过程中的剪枝



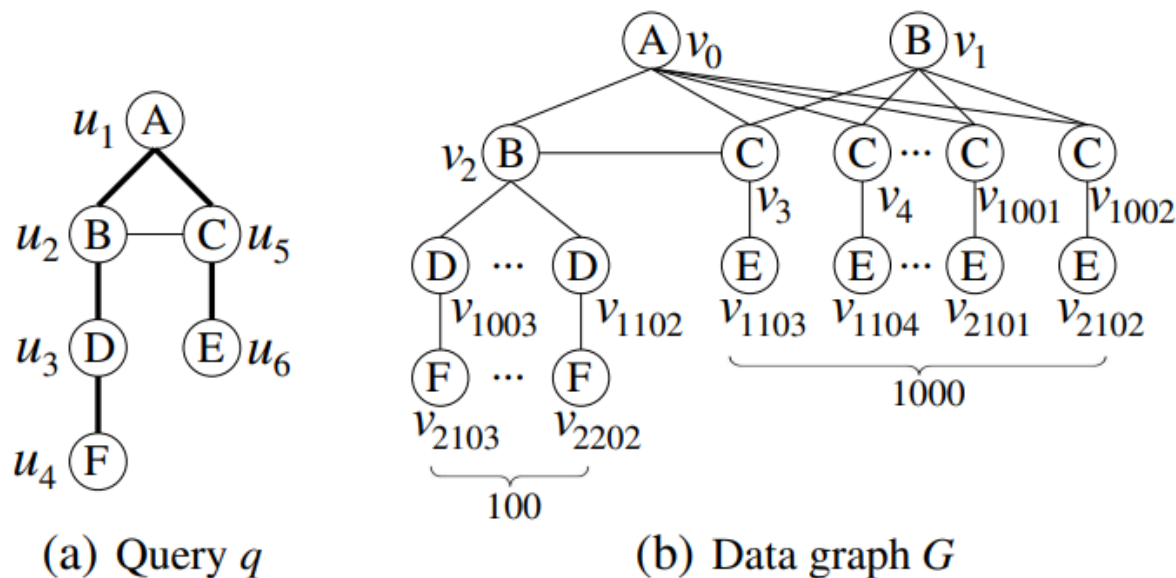
- ④搜索过程中的剪枝
 - 如全局索引的局部版本
 - 确定了查询、索引和匹配序之后，在传统技术框架中，搜索过程中的剪枝难以有更为突破的新方法

BGP 查询优化新技术主要来自于传统技术未重点覆盖的搜索过程

面向子图查询的Worst Case Optimal Join (WCOJ)

GraphFlow

Kankanamge et al. 2017



假设当前有一系列匹配了 u_1, u_2, u_3, u_4 的部分解，下一步需要扩展 u_5 ：

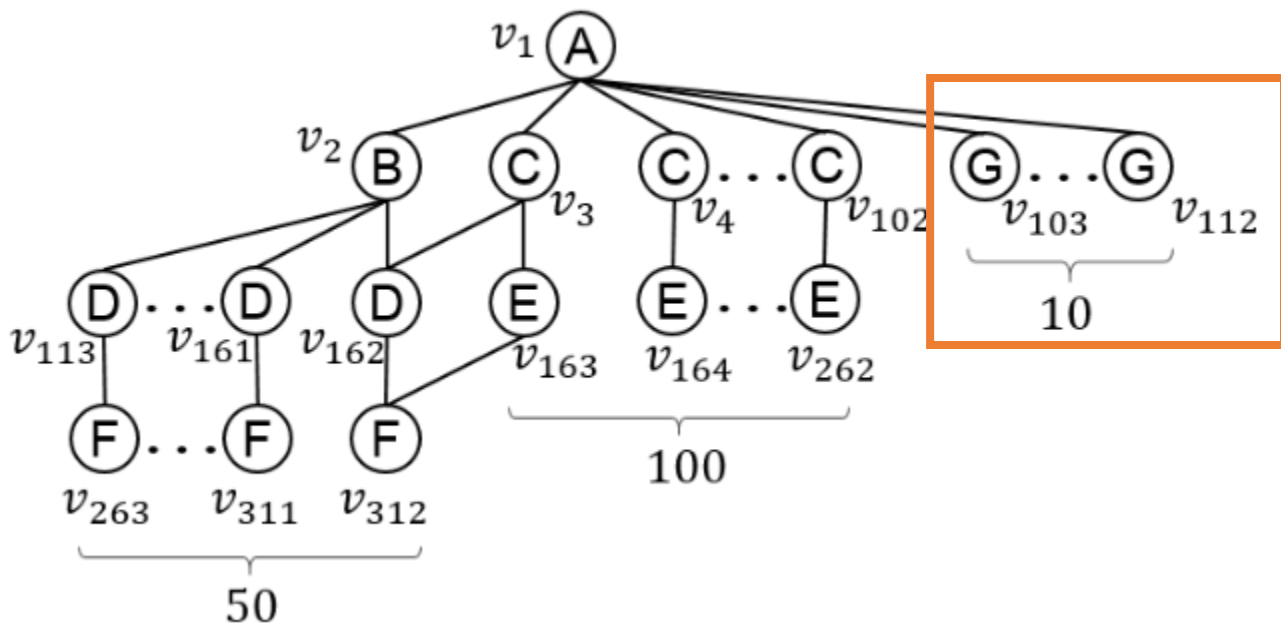
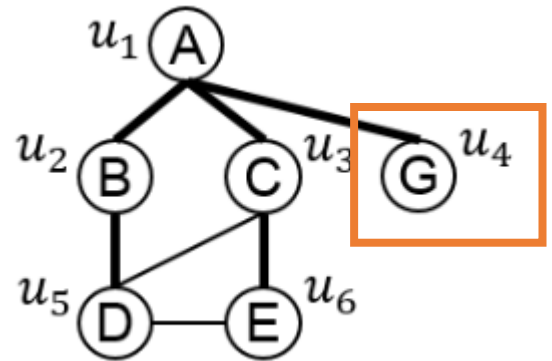
- 传统：先按 $v_3 \sim v_{1002}$ 进行联结（扩大1000倍），然后检查边 (u_2, u_5) 对应的数据边的存在（大幅减少）
- WCOJ：将 v_2 的邻居、 v_0 的邻居、 u_5 的候选进行交集，直接得到下一步扩展的分支，也附带处理了任何边检查（如， u_2 和 u_5 的边）

查询新技术：巧用搜索过程中的对称性之Failing Set

- 数据顶点在搜索过程中，发生“搜索失败”的对称性：
 - 如果两个顶点满足这个对称性，其中一个分支失败了，则对应的另一个分支也可以直接剪枝

DAF (DP-iso)

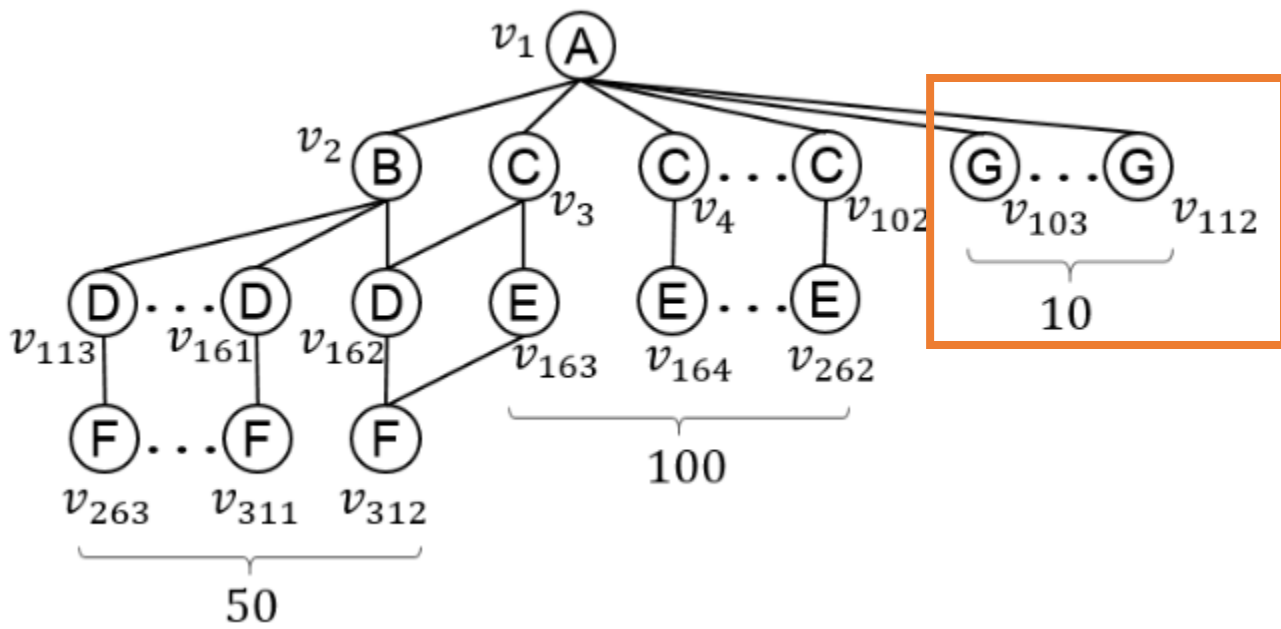
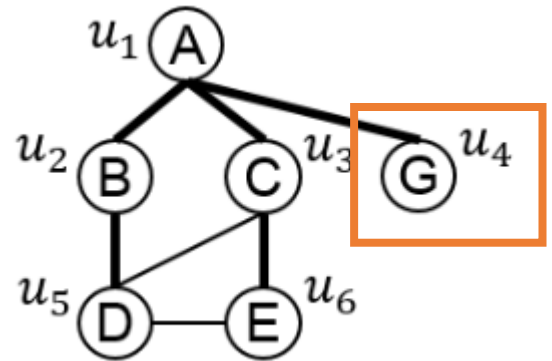
Han et al. 2019



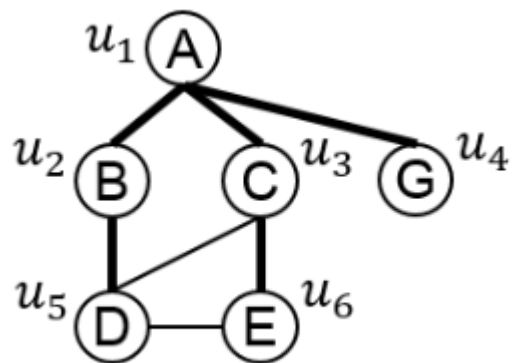
查询新技术：巧用搜索过程中的对称性之VEQ（顶点等价包括失败和成功）

- 数据顶点在搜索过程中，发生“搜索失败或成功”的对称性：
 - 除了failing set，还保存搜索成功后的结果集，遇到等价顶点之后，直接生成等价点（继续搜索将获得的）对应的结果

VEQ
Kim et al. 2021



查询新技术：局部笛卡尔积之下的交集缓存



$u_1 \rightarrow u_2 \rightarrow u_3 \rightarrow u_4 \rightarrow u_5 \rightarrow u_6$

$v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4$

v'_4

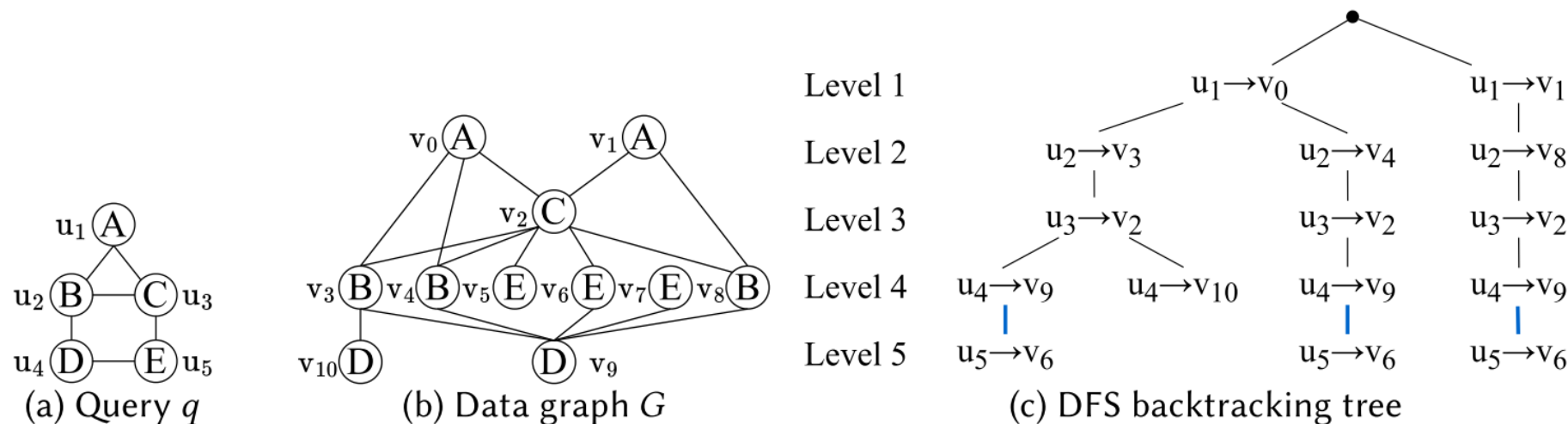
GraphFlow+

Mhedhbi et al. 2021

假设当前有一系列匹配了 u_1, u_2, u_3 的部分解 $\{v_1, v_2, v_3\}$,

- 扩展 u_4 的时候, 得到 n 个满足扩展需求的 v_4 ,
- 进而 n 个新的部分解, 而对这些部分解进行 u_5 扩展时, 交集操作完全一致, 即 v_2 邻居、 v_3 邻居和 u_5 候选的交集。
- 导致交集重复了 n 遍。
- 如果缓存交集结果, 可以通过复用来避免冗余的连续交集计算。

查询新技术：局部笛卡尔积之下的搜索压缩



Circinus

Jin et al. 2023

Circinus发现，相同的交集并不一定连续出现，可能间隔出现，需要压缩式回溯：

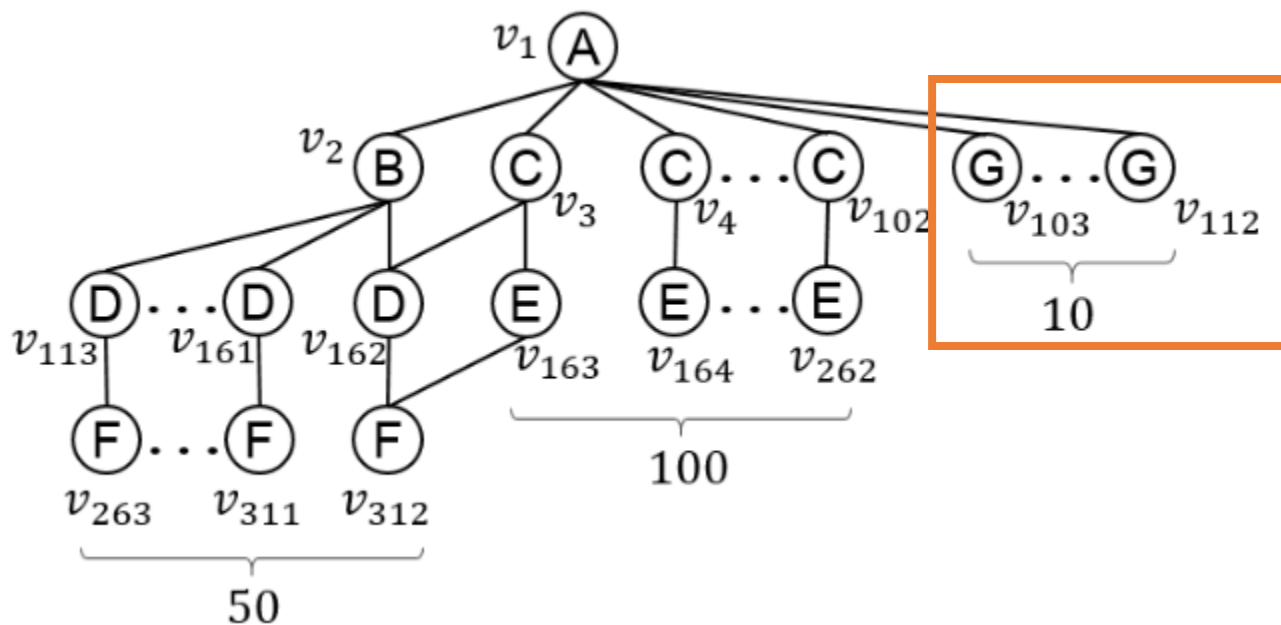
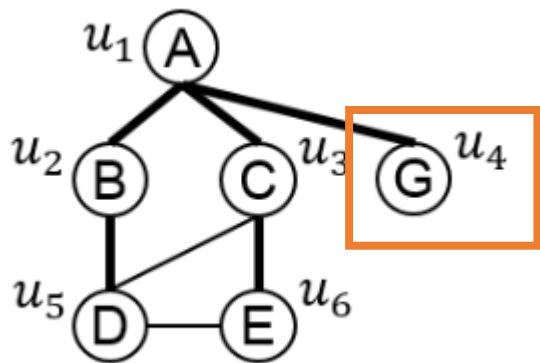
- 直接的方法是通过宽搜获得同一level的所有部分解，然后针对交集对部分解进行聚类，同类的共享交集计算，但空间容易指数爆炸
- 最优的办法是找到最优的子查询点集作为键值，映射同键值（如上例中的 $\{v_2, v_9\}$ ）的部分解为一组，共享交集
- Circinus并没提供最优的键值计算方法，通过启发式的手段计算一个查询点集的一个点覆盖作为键值，并没有确保达到最优情况

查询新技术：更细粒度的异步扩展初见端倪

- 隔离点：匹配序后方无邻居的查询点
 - 如下例，匹配序 $u_1, u_2, u_3, u_4, u_5, u_6$ 中的 u_4 ，因为 u_5, u_6 均不是 u_4 的邻居
- 推迟手段：
 - 将 u_4 直接推迟到最后进行扩展，不需要失败集或成功集，隔离点可能都多个，会形成独立集，没必要逐步扩展（造成连续笛卡尔积），而是将所有隔离点的候选在最后通过拼接的方式直接联结到部分解中，避免大幅扩大搜索空间。
 - 隔离点的候选点集如果是空集，可以提前剪枝

CaLiG

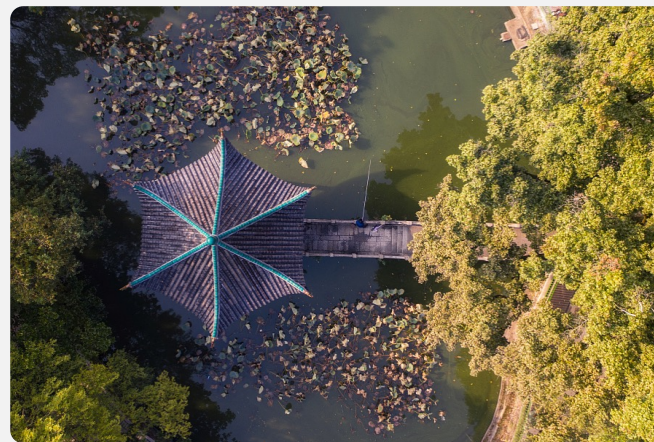
Yang et al. 2023



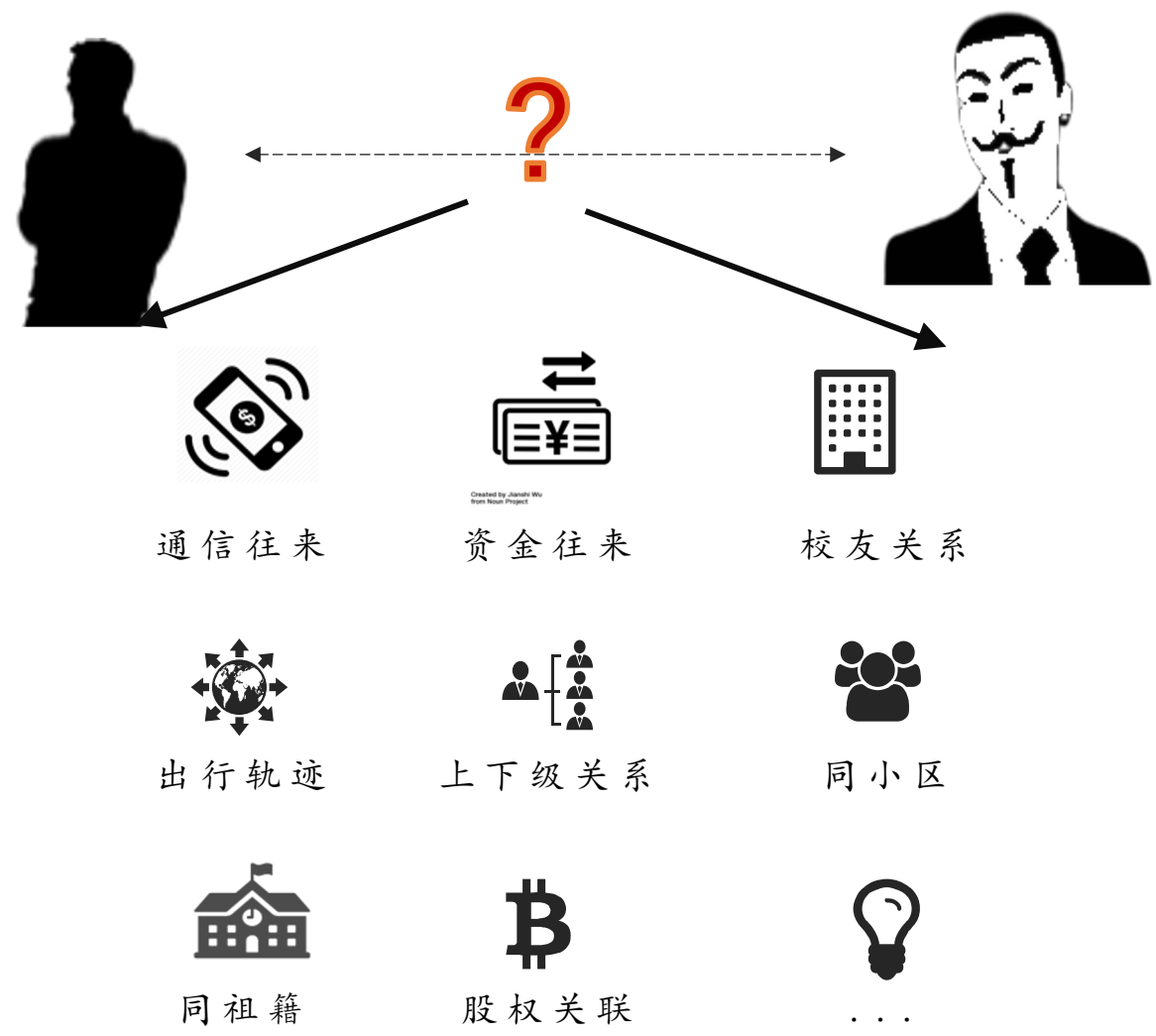
- 目前的查询**新技术**聚焦在搜索过程，相对传统技术在扩展（Extension）级别的粗粒度优化，新技术的优化策略优化粒度更细，包括
 - WCOJ
 - 失败集 （针对失败情况）
 - VEQ （成功集需要缓存结果，结果规模可能很大）
 - 交集缓存 （针对连续情况）
 - 压缩回溯 （无法达到最优情况）
 - 隔离点的特殊处理 （非隔离点的局部独立集无法处理好）

03 PART

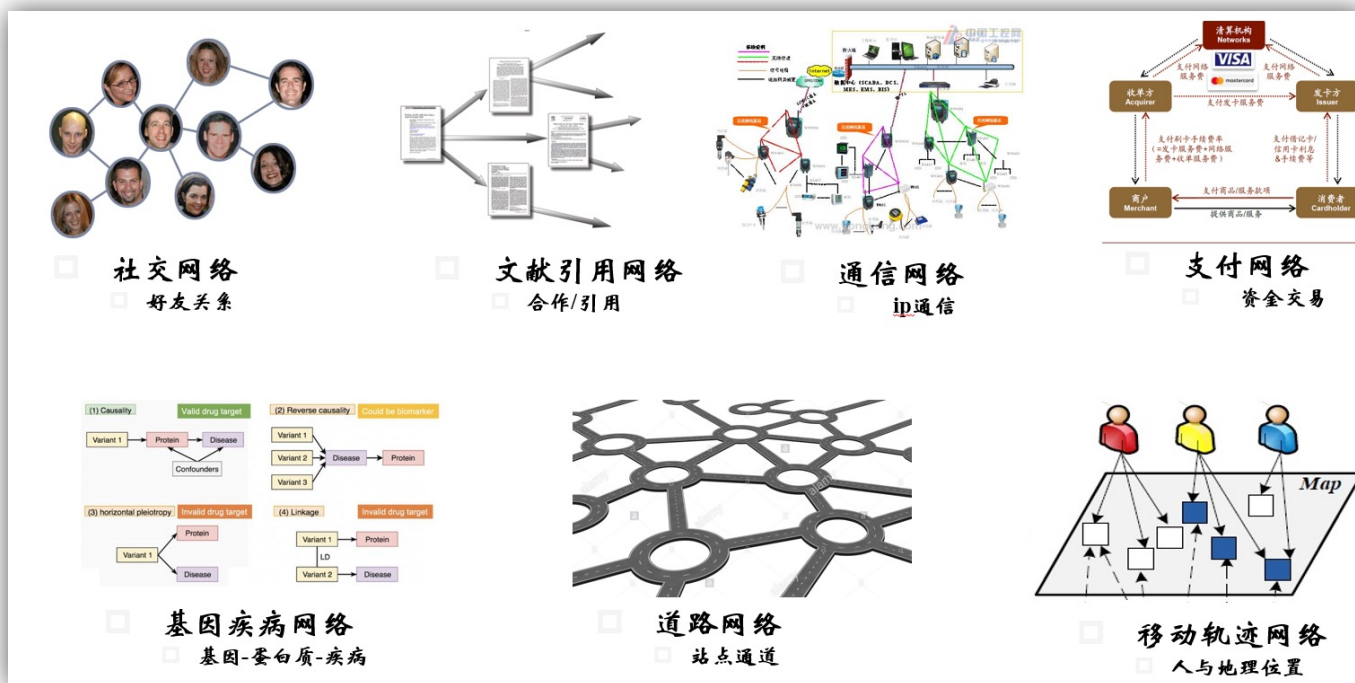
实体间关联查 询优化新技术



实体间关联查询优化新技术

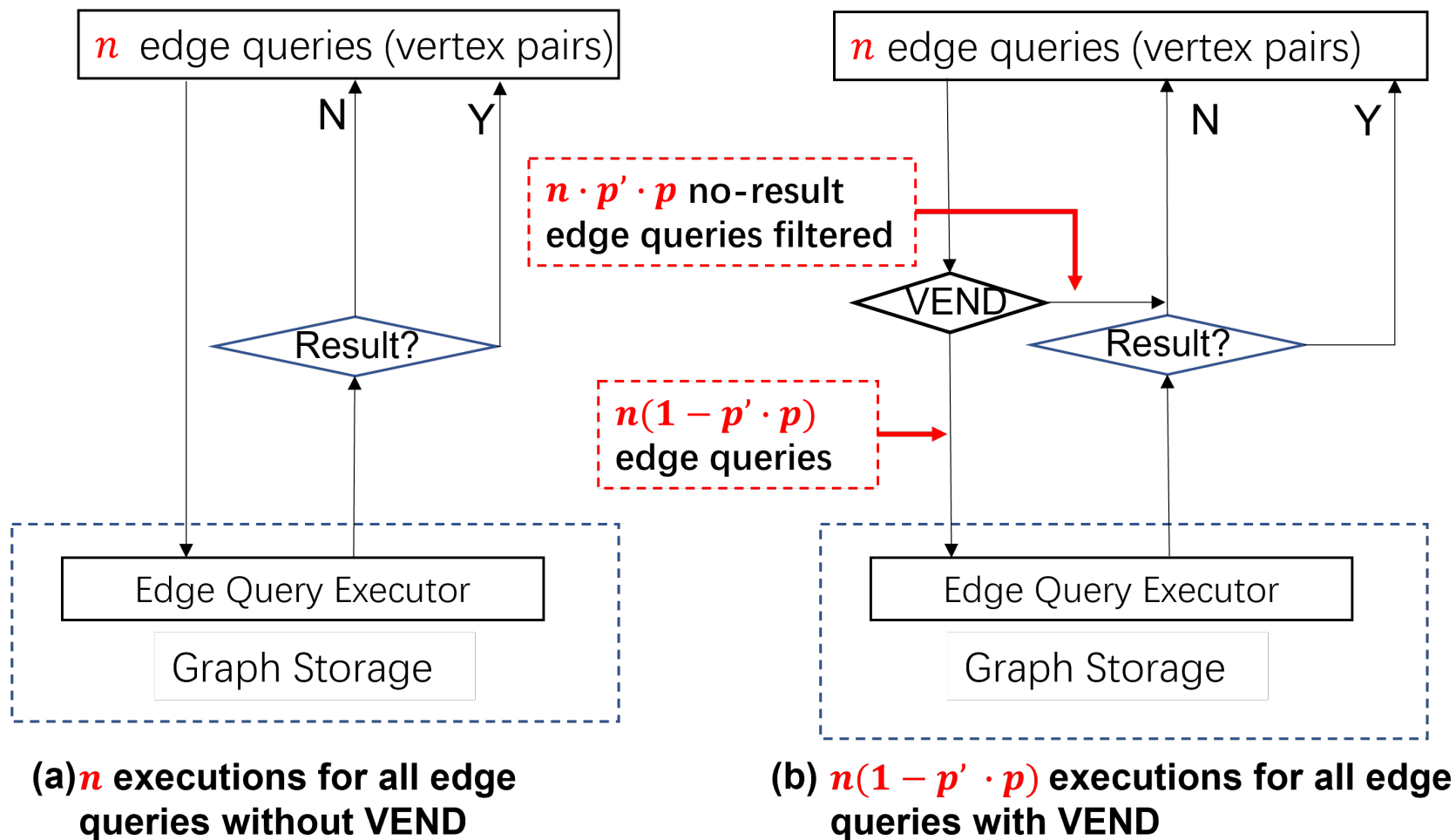


实体间关联查询优化新技术



➤ 现实图的常见特点

- $|V(G)| \gg \text{degr}(G)$
- 给定两点，有边概率为
 - $|E(G)| / (|V(G)| * (|V(G)| - 1) / 2)$
 - $= 2|E(G)| / (|V(G)| * (|V(G)| - 1))$
 - 即: $\text{degr}(G) / (|V(G)| - 1)$
 - 在百万边的大图上，这个概率能低于万分之一
- 关联查询，即边查询能有效返回的概率很低



Li et al. 2023
(ICDE)

VEND定义：编码函数和判定函数

Encoding Function f :

- Given a dimension number k , each vertex is assigned with **k-dimension** vector, denoted as $f(v)$.
- k is usually small**

Nepair $\text{NE}(G)$:

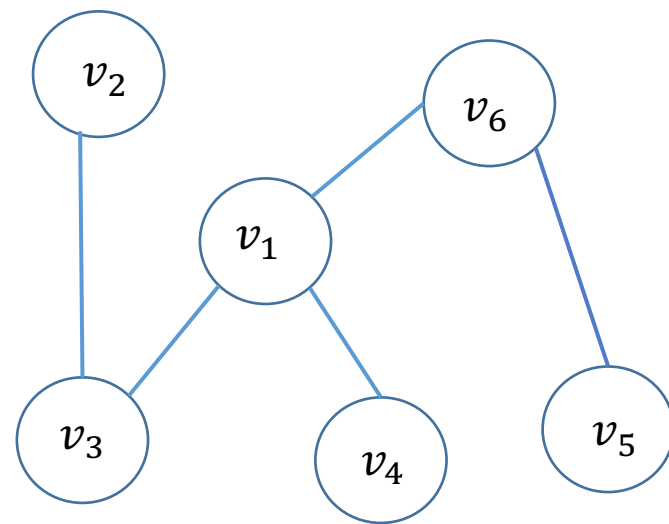
- given a graph $G = (V, E)$ and $v_1, v_2 \in G$, there is no edge connecting v_1, v_2 in G

$$\text{NE}(G) = \{(v_1, v_2) \mid v_1 \neq v_2 \wedge (v_1, v_2) \notin E\}$$

Nepair Determination Function F (NDF):

- A function to determine the NEpairs

$$F(f(v_1), f(v_2)) = 1$$



VEND打分和问题定义

VEND Score:

- The proportion of NEpairs that can be detected by F .

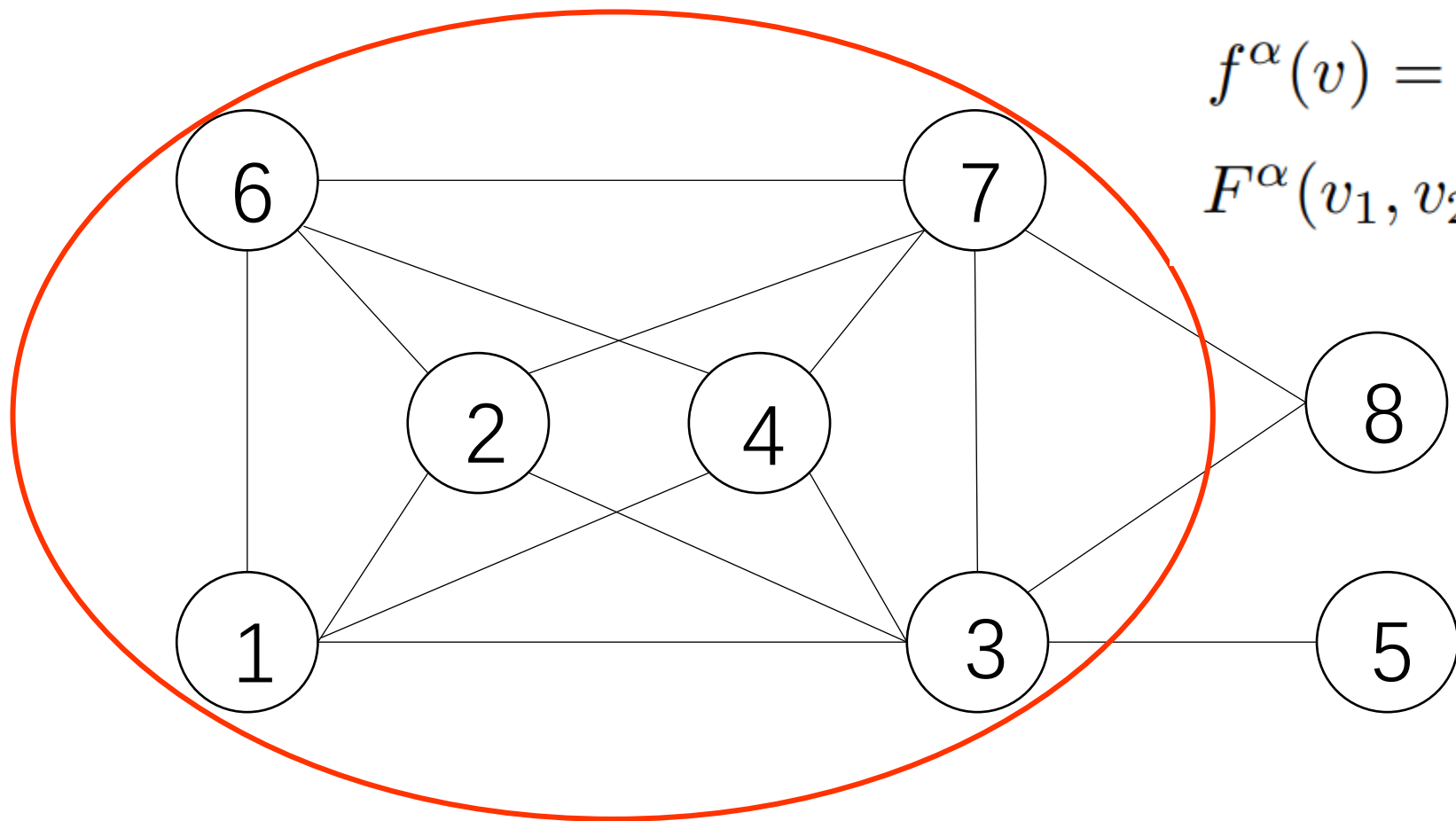
$$Score_{G,k}(f, F) = \frac{\sum_{v_1 \in V, v_2 \in V} (F(f(v_1), f(v_2)))}{|NE(G)|}$$

Problem Definition:

- Design an encoding function f and an NDF F
- $Score(f, F)$ is as high as possible
- f can be updated efficiently when edge updates

最优部分VEND: α 版本

Iteratively remove vertices with degree $< k$



$$f^{\alpha}(v) = [\tau_i, v_1, v_2, \dots, v_{|N_{G_i}(v)|}]$$

$$F^{\alpha}(v_1, v_2) = (v_2 \notin f^{\alpha}(v_1))$$

对于任意的数据图G，我们只需要考虑G的K-Core部分的编码，因为非K-Core部分总有完美策略

Figure: A data graph example and the core subgraph

基准策略：范围编码

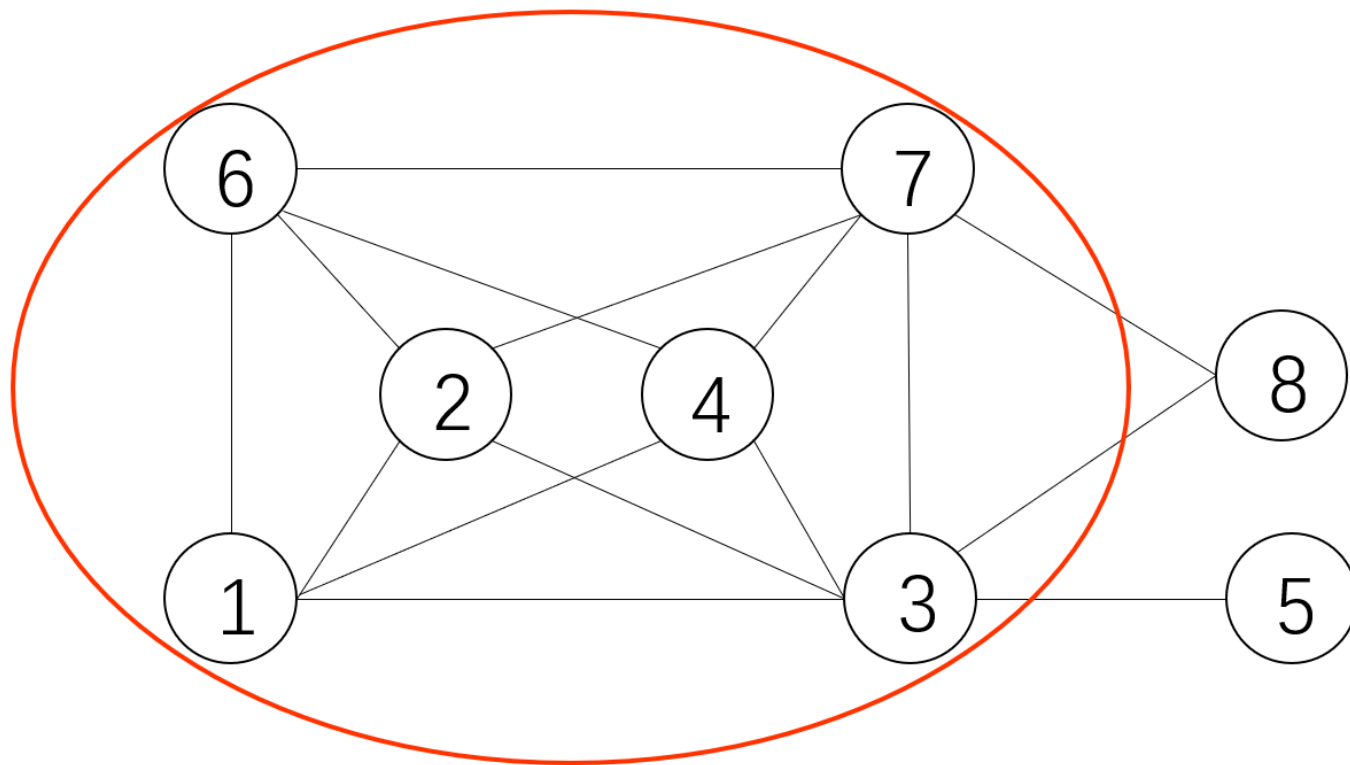
Range-based Encoding

Neighbor list after k-core: $N_{C_G^k}(v) = \{v_1, v_2, \dots, v_t\}$

- Choose the first k consecutive neighbor IDs (neighbor block) as encoding vector .
 - For any $v' \leq v_k$, v' is either an item in the vector (Neighbor), or v' and v are of no edge

基准策略：范围编码

Range-based Encoding



$k=3$

Leftmost: $f^\alpha(6) = \{1, 2, 4\}$

Rightmost: $f^\alpha(6) = \{2, 4, 7\}$

Range-based Encoding

Neighbor list after k-core: $N_{C_G^k}(v) = \{v_1, v_2, \dots, v_t\}$

- Choose the first k consecutive neighbor IDs (neighbor block) as encoding vector . ✗
- Choose the block with highest VEND score ✓

优化的范围编码

Optimized Range: Consider sequence $s = \{-\infty, v_1, v_2, \dots, v_{|N_G(v)|}, \infty\}$

不限定最左或最右的邻居块，而是可以选中间的某个

Basic Range

$$f^\alpha(1) = \{2, 3, 4\} \quad f^\alpha(4) = \{1, 3, 6\}$$

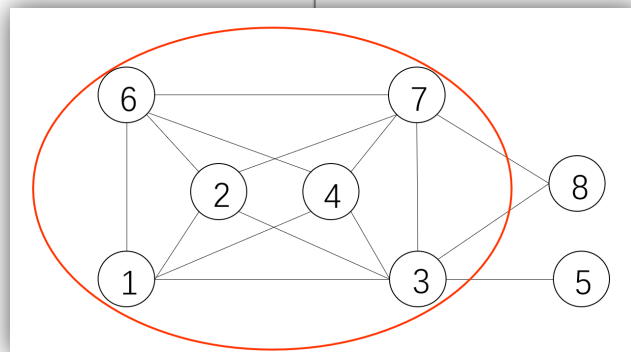
$$f^\alpha(2) = \{1, 3, 6\} \quad f^\alpha(6) = \{1, 2, 4\}$$

$$f^\alpha(3) = \{1, 2, 4\} \quad f^\alpha(7) = \{2, 3, 4\}$$



Detectable
NEpairs
within C_G^3

(2, 4)(4, 2)
(3, 6)(6, 3)



Improved Range

$$f^\alpha(1) = \{4, 6, \infty\} \quad f^\alpha(4) = \{1, 3, 6\}$$

$$f^\alpha(2) = \{1, 3, 6\} \quad f^\alpha(6) = \{2, 4, 7\}$$

$$f^\alpha(3) = \{4, 6, \infty\} \quad f^\alpha(7) = \{4, 6, \infty\}$$



(1, 7)(7, 1)
(2, 4)(4, 2)
(3, 6)(6, 3)

NEpair: 无边点对

Figure: Basic range VEND V.S optimized range VEND

Hash-based Encoding

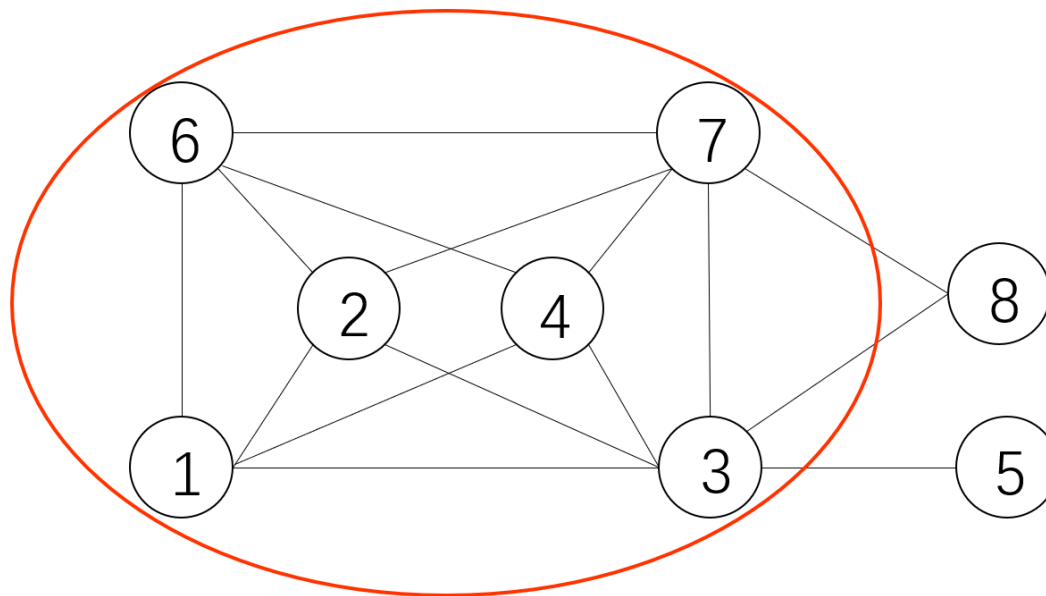
Neighbor list after k-core: $N_{C_G^k}(v) = \{v_1, v_2, \dots, v_t\}$

- Hash every neighbor IDs into a k-dimension vector
- Set the corresponding hash slot to 1.

$$f^{hash}(v)[v_1 \% k] = 1$$

混合编码：融合范围和哈希

一部分bit空间给范围编码，剩下的作为hash槽



An example for Hybrid VEND

$f^{hyb}(6)$: ● Set {4,7} for range , remaining {1,2} for hash

NDF: ● Determine if the target within the range, otherwise examine the hash

混合编码的关键：最优编码选择

incorporating range and hash based ideas

- Encoding with k' dimension for range
- Remaining $k - k'$ for hash

How to determine the best range block ?

- Set VEND score as metric
- Dynamic block selection

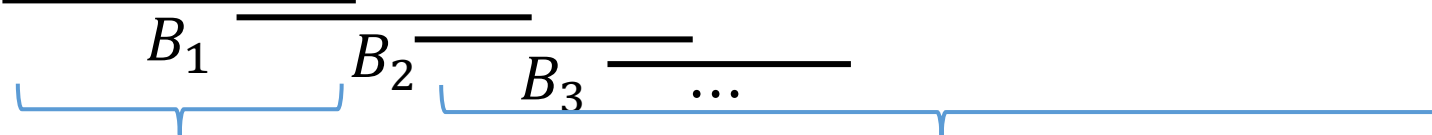
基于滑动窗口的动态块选取策略

Dynamic Block Selection $1 \leq |B| \leq k$

- Consider all t-size blocks
- Sliding-window approach (avoid duplicate computation)

When $|B| = 2$

$$N_G(v) = \{-\infty, v_1, v_2, v_3, v_4, v_5, v_6, \dots, v_t, \infty\}$$



For Range

For Hash

Consider $|B| = 3, 4, \dots, k$

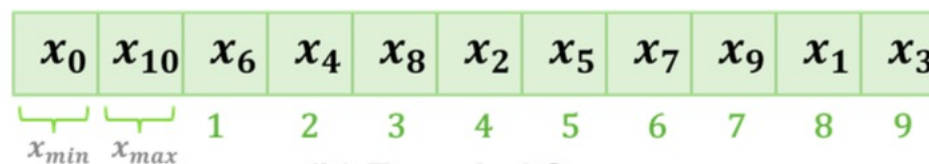
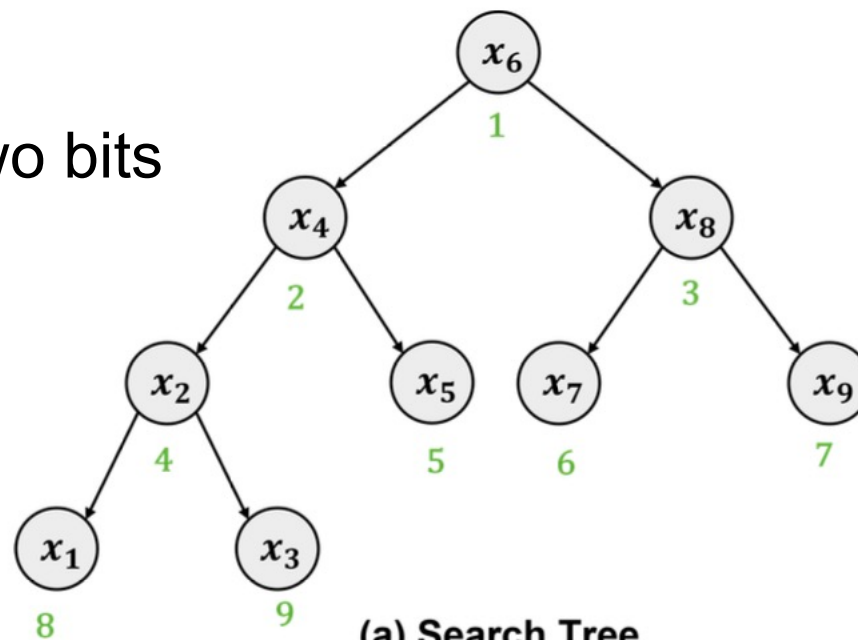
VEND score

Time Complexity : $O(k^2 \cdot |N_G(v)| \cdot I)$

$I = 32$

Optimizations

- Encoding compression
- Indicating infinite flags with only two bits
- SIMD



Efficient Maintenance

实验：设置

- CPU : Intel(R) Xeon(R) Silver-4210R 2.40GHz
- Main memory :128G
- C++

Datasets	Vertexs	Edges	Degrees
As-Skitter	1,696,415	11,095,298	13
Wiki-topcats	1,791,489	25,444,207	28
Uk-2005	39,454,463	783,027,125	40
Gsh-2015	988,490,691	25,690,705,118	52
Orkut	3,072,441	117,185,083	76

Table: Summary of Datasets

实验：对比工作

Methods:

- SBF : Standard Bloom Filter
- CBF : Counting Bloom Filter
- BBF : Blocked Bloom Filter
- LBF : Local Bloom Filter
- Range : Range-based VEND (our baseline)
- Hybrid : Hybrid VEND (our final solution)

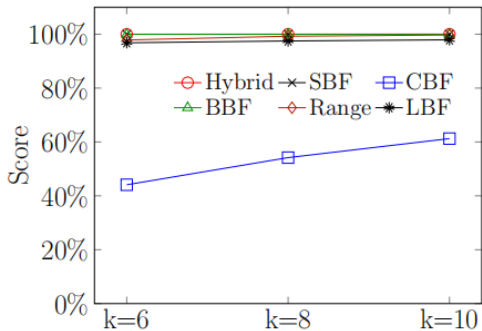
Evaluation:

- VEND score
- Edge queries acceleration
- Maintenance efficiency

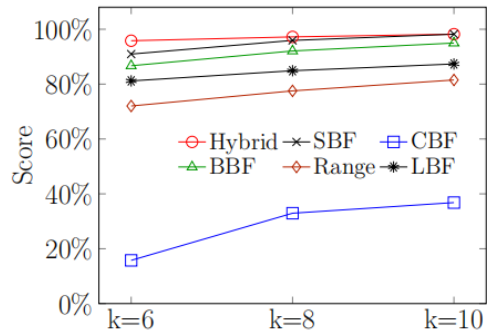
$k = 6, 8, 10$

随机点对的VEND分

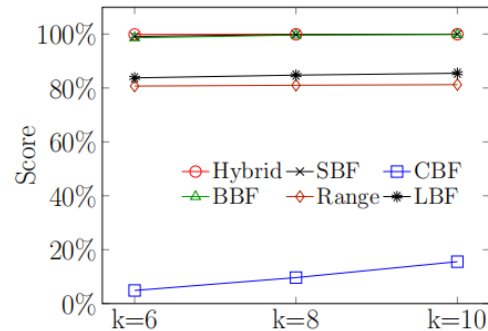
我们的版本 > 98%，即能过滤98%的无边顶点对，大幅加速随机边查询



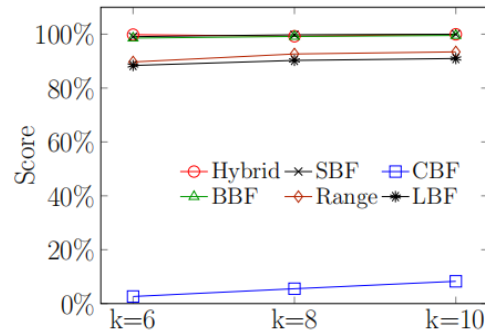
(a) As-Skitter



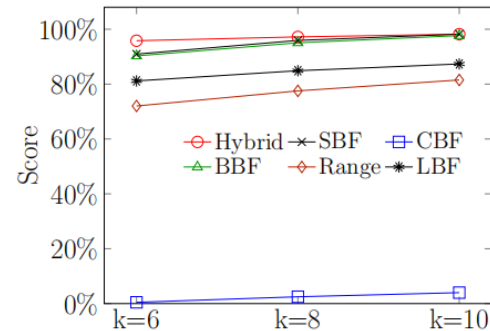
(b) Wiki-topcats



(c) Uk-2005



(d) Gsh-2015



(e) Orkut

Figure: VEND Score over Different Datasets on Randomly Generated Vertex Pairs

共同邻居点对的VEND分

性能弱于随机点对，符合预期，优化仍然可观

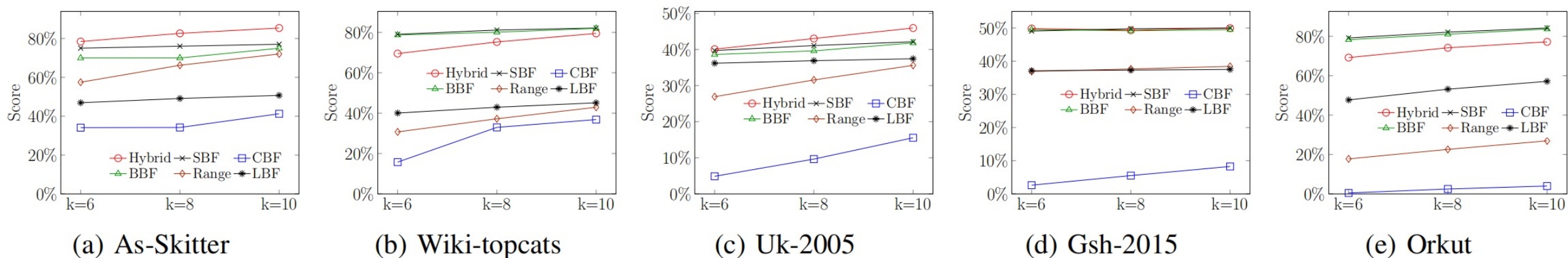


Figure: VEND Score over Different Datasets on Vertex Pairs of Common Neighbor

实测边查询加速

VEND加速的边查询具有将近两个数量级的优势

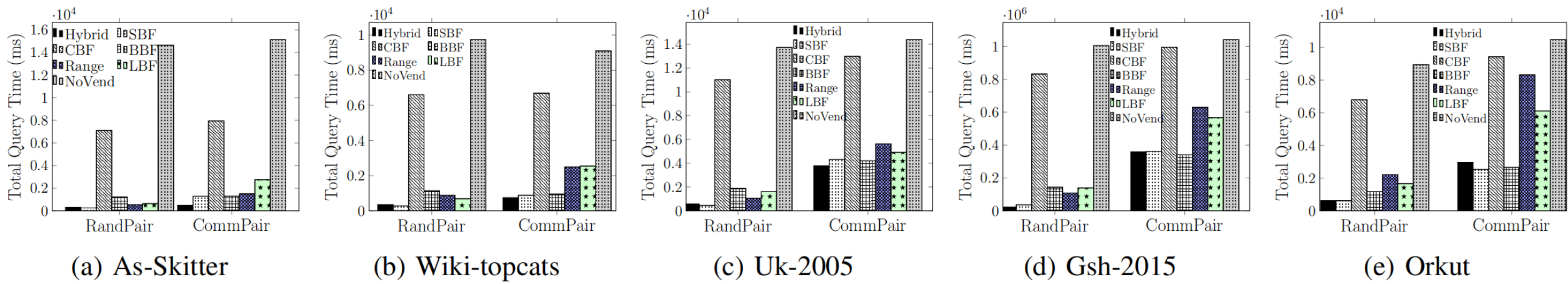


Figure: Total Time over Different Edge Query Sets (k = 8)

- We propose a new and important problem for determining no-result edge queries (VEND).
- We formally define an important measure (VEND source) to precisely evaluate the performance of VEND solutions
- We propose an effective VEND solution and design efficient update algorithms over the hybrid version.
- Various experiments indicate that our VEND solution can explicitly improve the corresponding system performance

04 PART

知识图谱查询优 化新技术总结



- 知识图谱的核心查询—**BGP**查询的传统优化主要在查询重写、索引设计和匹配序生成方面，而搜索过程中的新加速策略成为了目前主流的优化新技术着力点，尤其在去除冗余计算方面的细粒度策略
- 点编码加速实体间关联查询（**VEND**）是一个全新的问题，最为主流图谱系统中的基础操作（边查询），**VEND**具有重要的意义

2023

Thank you~