

DP-ISO说明文档

概述

DP-ISO是基于本组对于图匹配的接口需求以及效率需求设计的基于DAF思想的子图同构算法

算法概述

DP-ISO的主要优化部分分为以下三个部分

DAGDP

对于query的每个弱连通子图 构建一个DAG并根据邻接关系减少candidate set的规模

Adaptive Matching Order

采用了candidate size order作为match order，同时采用动态更新candidate set的方式

Fail Set

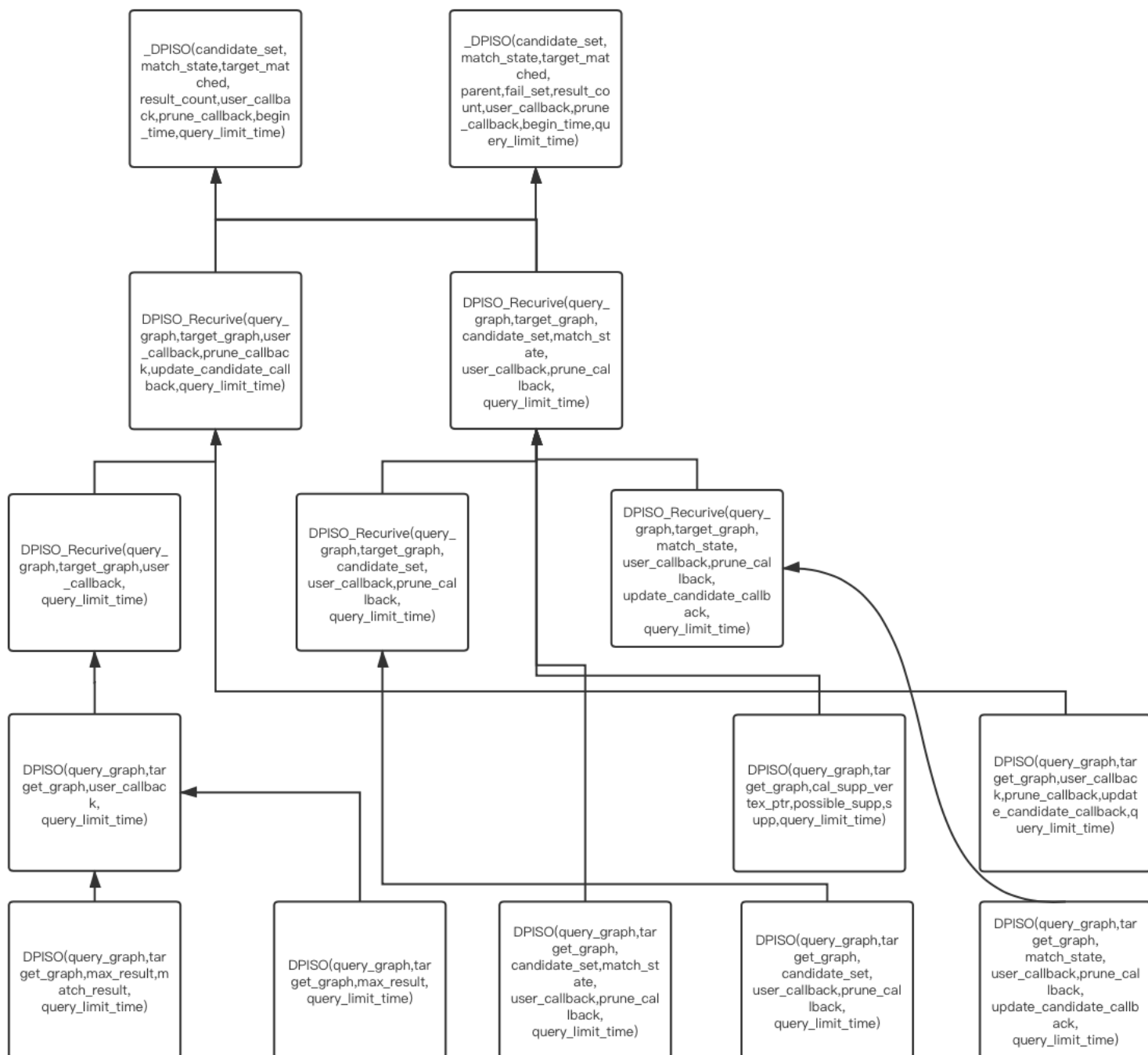
利用匹配时的先验知识对当前状态进行剪枝，从而减少搜索空间

与DAF的不同

- 1.不采用拓扑序而采用邻接节点，可以支持部分匹配
- 2.考虑到Graph相关的遍历性能，不构建candidate space
- 3.增加了匹配的自适应能力，若query较小则不采用fail set优化
- 4.可匹配带有多个连通块的query

代码设计

箭头表示调用关系



可用接口及其相关参数说明

constexpr size_t large_query_edge = 6

若query的边数超过这个数目，则采用fail set

**DPISO(const QueryGraph &query_graph, const TargetGraph
&target_graph, MatchCallback user_callback, double query_limit_time = 1200)**

给定Query Graph和Target Graph的情况下，通过user_callback进行匹配结果的处理，默认设定的执行时间为1200s（若超过这个时间，则强制结束）

DPIISO(const QueryGraph &query_graph, const TargetGraph &target_graph, int max_result, double query_limit_time = 1200)

给定Query Graph和Target Graph的情况下，计算匹配数量（不超过max_result），默认设定的执行时间为1200s（若超过这个时间，则强制结束）

int DPIISO(const QueryGraph &query_graph, const TargetGraph &target_graph, int max_result, ResultContainer &match_result, double query_limit_time = 1200)

给定Query Graph和Target Graph的情况下，计算匹配数量（不超过max_result）并将其存储在match_result中，默认设定的执行时间为1200s（若超过这个时间，则强制结束）

int DPIISO(const QueryGraph &query_graph, const TargetGraph &target_graph, typename QueryGraph::VertexConstPtr cal_supp_vertex_ptr, const std::vector &possible_supp, std::vector &supp, double single_query_limit_time = 100)

给定Query Graph和Target Graph的情况下，根据possible_supp的结果计算匹cal_supp_vertex_ptr的supp并将其存储在supp中，默认possible_supp内每个点的匹配计算时间为100s（若超过这个时间，则认为该点不为supp）

int DPIISO(const QueryGraph &query_graph, const TargetGraph &target_graph, MatchCallback match_callback, PruneCallback prune_callback, UpdateInitCandidateCallback update_initcandidate_callback, double query_limit_time = 1200)

给定Query Graph和Target Graph的情况下，通过user_callback进行匹配结果的处理，通过prune_callback进行剪枝处理，通过update_initcandidate_callback来人工对candidate_set进行一定的限制。默认设定的执行时间为1200s（若超过这个时间，则强制结束）

int DPIISO(const QueryGraph &query_graph, const TargetGraph &target_graph, std::map> &candidate_set, std::map &match_state, MatchCallback user_callback, PruneCallback prune_callback, double query_limit_time = 1200)

在已知candidate_set和match_state的情况下，给定Query Graph和Target Graph，通过user_callback进行匹配结果的处理，通过prune_callback进行剪枝处理。默认设定的执行时间为1200s（若超过这个时间，则强制结束）

tips: candidate_set和match_state在运行时会被修改，外部做好备份

int DPIISO(const QueryGraph &query_graph, const TargetGraph &target_graph, std::map> &candidate_set, MatchCallback user_callback, PruneCallback prune_callback, double query_limit_time = 1200)

在已知candidate_set的情况下，给定Query Graph和Target Graph，通过user_callback进行匹配结果的处理，通过prune_callback进行剪枝处理。默认设定的执行时间为1200s（若超过这个时间，则强制结束）

tips:candidate_set在运行时会被修改，外部做好备份

```
int DPISO(const QueryGraph &query_graph, const TargetGraph &target_graph, std::map  
&match_state, MatchCallback user_callback, PruneCallback  
prune_callback, UpdateCandidateCallback update_candidate_callback, double  
query_limit_time = 1200)
```

在已知match_state的情况下，给定Query Graph和Target Graph，通过user_callback进行匹配结果的处理，通过prune_callback进行剪枝处理,通过update_initcandidate_callback来人工对candidate_set进行一定的限制。。默认设定的执行时间为1200s（若超过这个时间，则强制结束）

tips:match_state在运行时会被修改，外部做好备份