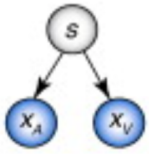
به نام خدا

علی فتحی ۹۴۱۰۹۲۰۵

تمرین کامپیوتری سری ۳ استنتاج علّی

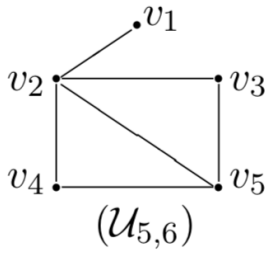


بخش اول: پیاده‌سازی تابع محاسبه سایز MEC

کد این بخش در توابع موجود در MEC.py در فولدر تمرین آمده است. توضیح مختصر توابع به صورت زیر است:

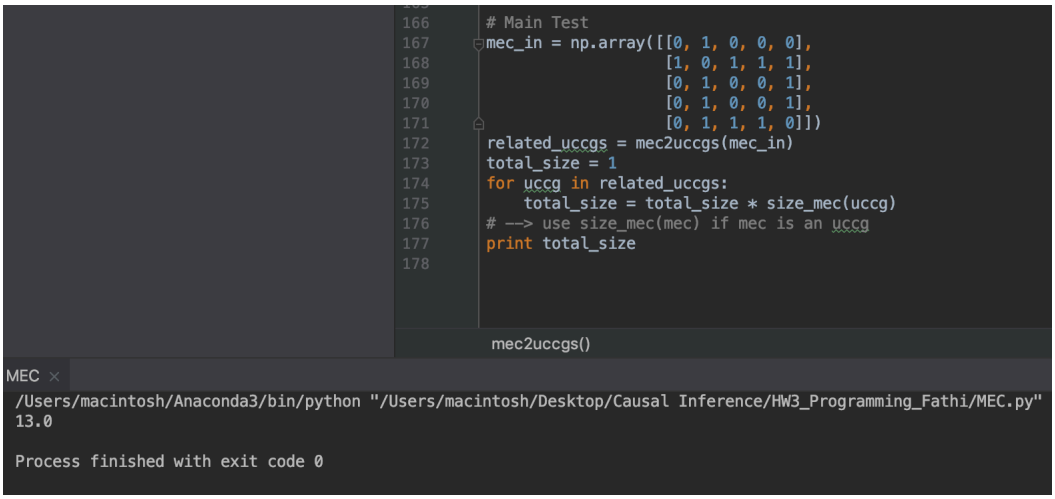
نام تابع	عملکرد
adjacent_orienting	مجموعه T را پیدا می‌کند و از A به آن فلش جهت‌دار می‌کشد.
meek_orient	با قوانین میک، یال‌های G_T را جهت‌دار می‌کند.
is_neighbour	بررسی می‌کند دو یال، راس مشترکی دارند یا خیر
place_of_one	اندیس یکی زوج از یال‌هایی با راس مشترک را برمی‌گرداند
mix_neighbours	یال‌های بی‌جهت می‌گیرد و آنها را ادغام کرده و رئوس uccg های متناظر را برمی‌گرداند
nodes2uccg	رِئوس یک uccg را می‌گیرد و با توجه به گراف اولیه ماتریس مجاورت uccg آن را برمی‌گرداند
undirected	یال‌های بی‌جهت یک گراف را جدا کرده و به کمک توابع بالا، uccg های آن را برمی‌گرداند.
chain_com	تابع الگوریتم ۱، uccg های جهت دهی شده با راس v را برمیگرداند.
size_mec	تابع الگوریتم ۲، تعداد گراف‌های هم ارزی مارکوف را برمی‌گرداند
mec2uccgs	ورودی را اگر uccg نباشد، به uccg هایی تقسیم می‌کند

نمونه یک خروجی برای گراف زیر نیز به صورت زیر است (مثال و نتیجه از مقاله اصلی آورده شده است):



$(\mathcal{U}_{5,6})$

$$\text{SizeMEC}(\mathcal{U}_{5,6}) = 13$$



که به عنوان تست، تعداد ۱۳ را به درستی محاسبه کرده است.

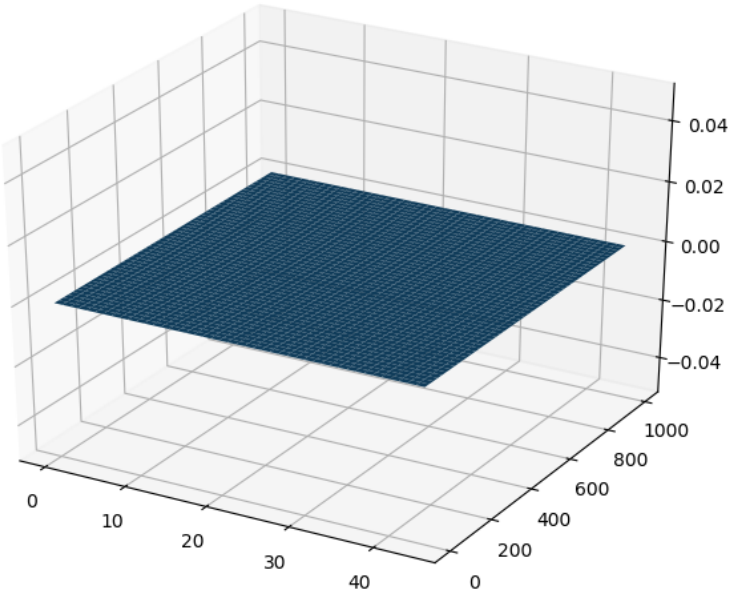
بخش دوم: محاسبه توزیع اندازه MEC و تعداد یال‌ها

در همان MEC.py توابع زیر به کد اضافه شده اند:

نام تابع	عملکرد
markov_chain	زنجیر مارکوف را شبیه سازی می‌کند
to_next_state	با انتخاب یک action مجاز به حالت بعد در زنجیر می‌رود
set_of_valid_operations	تمام action های مجاز در یک حالت را برمی‌گرداند
is_remove_v_valid	بررسی می‌کند عمل حذف v-structure مجاز است یا خیر.
is_insert_d_valid	بررسی می‌کند عمل قرار دادن یال جهت دار مجاز است یا خیر
is_insert_u_valid	بررسی می‌کند عمل قرار دادن یال بی جهت مجاز است یا خی
does_path_cross	بررسی میکند تمام مسیرها از یک گره به گره دیگر توسط مجموعه‌ای block می‌شود یا خیر
parents	parent ها را برمی‌گرداند
neighbours	همسایه‌ها را برمیگرداند
is_clique	بررسی میکند یک زیر گراف، کلیک هست یا خیر
various_pairs	تمام یال‌ها و v-structure را برمیگرداند
pdag_initializer	یک pdag اولیه میسازد (برای شروع زنجیره مارکوف)

اما کد این بخش، در تابع does_path_cross کامل نشده است، و سوالی در این باره در پیاتزای درس مطرح کردم اما بعد از یک هفته پاسخی به آن داده نشد!!

نتیجه یک اجرا با وجود نقص در یک تابع به شکل زیر است:

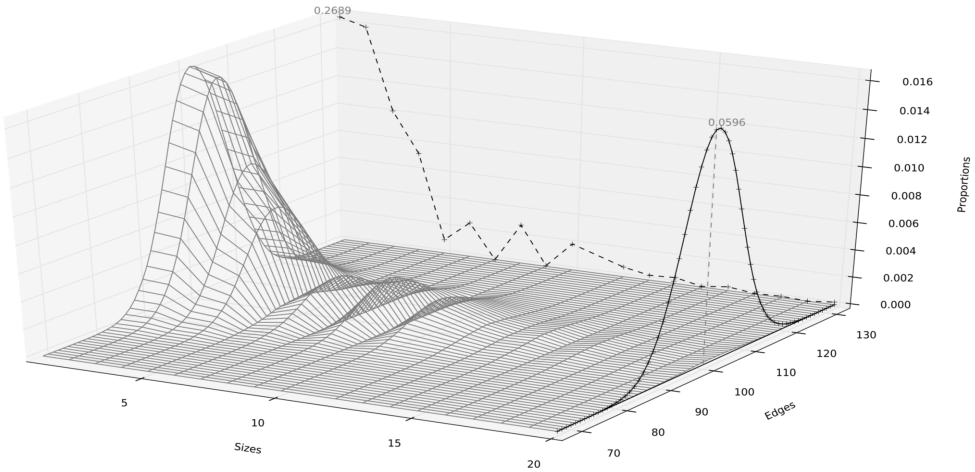


لازم به ذکر است پیاده سازی Markov Chain در این بخش کامل می‌باشد و تنها شروط بی شرط زیر بررسی نشده اند:

Validity Tests

(1) Every undirected path from x to y contains a node in $N_{x,y}$

و نتیجه درست هم بعد از پیاده سازی کامل باید همچین شکلی می‌شد:



❊ ❊ ❊