

project ۲.۲ AI

۹۷۱۲۰۴۶ - شیرین محمدی

۲۰۲۲ January

۱ طرح مسئله

یک شبکه ساده و ارتباطات آن می تواند بصورت یک گراف بدون جهت و بدون وزن توصیف شود. همچنین ماتریس مجاورت راه مناسبی برای دسترسی به اجتماعات مختلف این شبکه می تواند باشد. حال نیازمند مقیاسی جهت ارزیابی اجتماع شناسایی شده هستیم. در الگوریتم Girvan Newman مفهومی به نام modularity function بصورت زیر تعریف شد:

$$Q = \sum_{u,v} [A_{u,v} - \frac{k_u * k_v}{2\xi}] \delta(u, v)$$

که در اینجا $A_{u,v}$ مقادیر u و v در ماتریس مجاورت هستند (اگر گراف وزن دار باشد، برابر با وزن یال خواهد بود) همچنین از آنجاییکه گراف وزن دار نمی باشد ξ برابر با تعداد یال های گراف می باشد. k_u و k_v هم مجموع اتصالات موثر به دو گره u و v می باشد.

۲ الگوریتم بهینه سازی فاخته

در این الگوریتم هرکدام از اجتماعات مورد بررسی به عنوان یک شئی به نام cuckoo معرفی می شوند.

۱.۲ مقدار دهی اولیه

همانگونه که در الگوریتم ژنتیک کروموزوم ها بیانگر پاسخ نهایی که باید مشخص شده و در ابتدا مقدار دهی می شدند، در این الگوریتم habitat مشابه آن می باشد. habitat بیانگر موقعیت فعلی زندگی کوکو ها می باشد. در اینجا اگر habitat در جایگاه i دارای مقدار j باشد به این معنی است که دو نود i و j امین همسایه اش در یک اجتماع قرار می گیرند علاوه بر این مفهومی تحت عنوان ELR معرفی میشود که بیانگر ناحیه ای است که کوکو در آن تخم میگذارد.

$$ELR = \alpha * \frac{\text{numberofcurrentcuckoo'seggs}}{\text{totalnumberofeggs}} * (var_{hi} - var_{low})$$

۲.۲ تابع هدف

هر شئی کوکو دارای یک تابع Q برای ارزیابی مناسب بودن محل زندگی دارد که در اینجا جهت شناسایی اجتماعات مناسب به کار می رود و همان modularity function ذکر شده در بخش طرح مسئله می باشد.

حال ما در کد ارائه شده هربار Q را برای کوکو های بدست آمده محاسبه می کنیم و ماکسیمم آن ها را جهت بخش بندی اجتماعمان استفاده می کنیم. برای محاسبه Q برای فاخته مان از کلاس DS که همان ساختمان داده disjoint set است، استفاده می کنیم.

۳.۲ spawn

در این تابع با احتمال مشخصی تغییر در habitat های تخم های جدید ایجاد شده، اعمال می شود و در نهایت قرار گیری در ناحیه ELR بررسی می شود. در اینجا از تابع diff که اختلاف دو کوکو را بر حسب اختلاف habitat شان محاسبه می کند، استفاده می شود.

۴.۲ best egg

در این قسمت با استفاده از تابع merge sort تخم هایمان را براساس Q شان مرتب می کنیم. سپس تخم هاییکه بهترین Q در راستای حل تابع هدف ما را دارند انتخاب می کنیم تا زمانیکه تعداد تخم ها برابر با جمعیت تعیین شده باشد.

۵.۲ migration

در اینجا از و تابع، یکی جهت پیدا کردن مقصد و دیگری برای پیاده سازی، استفاده می کنیم. در تابع cor migration از K Means clustering جهت تقسیم بندی کوکو ها به سه قسمت استفاده می شود و در نهایت بخش با کوکو های دارای بشتترین Q انتخاب می شوند. پس از بدست آوردن مرکز بخش انتخاب شده به عنوان مقصد مهاجرت و تعریف کردن habitat آن، با اجرای تابع migration اعضای habitat کوکو با توجه به فاصله ای که با مقصد دارند در habitat مقصد جایگیری میکند و habitat جدید شکل می گیرد.

۶.۲ پایان الگوریتم

در نهایت پس از پایان یافتن تکرار ها آخرین لیست تخم ها تبدیل به کوکو شده و هر کوکویی که بیشترین Q را داشته باشد انتخاب می شود.