درس تحقیق در عملیات پیشرفته

مدرس: دکتر مسعود یقینی

تمرین شماره ۲ : حل مساله TSP با استفاده از الگوریتم ACS

تاریخ تهیه: ۸۸/۱۰/۱۰

شرح تمرین:

هدف تمرین طراحی و پیاده سازی یک الگوریتم کاراً (با زمان حل قابل قبول)، موثر (ارائه جوابهای نزدیک به جواب بهینه) و قوی (robust) بر اساس الگوریتم ACS است.

برای ارزیابی عملکرد الگوریتم طراحی شده ۱۲ مساله تست از سایت TSPLIB همراه با تمرین بر روی سایت گذاشته شده است. این مسائل از اندازه های کوچک شروع و به صورت صعودی اندازه آنها بزرگتر می شود.

Row	Problem	Distance	No. of Nodes	Optimum value
1	ulysses16	Geographic	16	6859
2	ulysses22	Geographic	22	7013
3	Eil51	Euclidean	51	426
4	Berlin52	Euclidean	52	7542
5	kroA100	Euclidean	100	21282
6	rd100	Euclidean	100	7910
7	gr202	Geographic	202	40160
8	a280	Euclidean	280	2579
9	pcb442	Euclidean	442	50778
10	gr666	Geographic	666	294358
11	pr1002	Euclidean	1002	259045
12	u1060	Euclidean	1060	224094

لازم به ذکر است که برای هر نقطه (شهر) در مسائل تست مختصات آنها داده شده است. مختصات به دو صورت **مختصات فضای دو بعدی** و یا **مختصات جغرافیایی** است. در داخل فایلهایی که مسائل آن با متخصات در فضای دو بعدی داده شده است بصورت EDGE_WEIGHT_TYPE : EUC_2D و در مسائلی که مختصات جغرافیایی یعنی طول و عرض جغرافیایی داده شده است، بصورت EDGE_WEIGHT_TYPE: GEO مشخص گردیده است.

لازم به ذکر است که بایستی فواصل شهرها ابتدا محاسبه شده و در یک فایل ذخیره شده و سپس برای اجرای الگوریتم از فایل مربوط به فواصل استفاده شود. زیرا اگر هر بار فواصل محاسبه شود، زمان الگوریتم بصورت کاذب افزایش پیدا می کند.

روشهای محاسبه فواصل:

محاسبه فاصله اقلیدسی:

$$d(i,j)=\sqrt{(|x_1-x_2|^2+|y_1-y_2|^2)}$$

محاسبه فاصله جغرافیایی:

فرض کنید x[i] و y[i] طول و عرض جغرافیایی داده شده نقطه i در مساله است. ابتدا باید کلیه مقادیر داده شده را با استفاده از روابط زیر به واحد رادیان تبدیل کنیم:

```
PI = 3.141592 ;
deg = int( x[i] );
min = x[i] - deg ;
latitude[i] = PI * (deg + 5.0 * min / 3.0 ) / 180.0 ;

deg = int( y[i] );
min = y[i] - deg ;
longitude[i] = PI * (deg + 5.0 * min / 3.0 ) / 180.0;
```

پس از محاسبه طور و عرض جغرافیایی بر حسب رادیان، با استفاده از روابط زیر می توانیم فاصله دو نقطه i و j را بر حسب کیلومتر محاسبه کنیم:

```
RRR = 6378.388 ;
q1 = cos( longitude[i] - longitude[j]);
q2 = cos( latitude[i] - latitude[j]);
q3 = cos( latitude[i] + latitude[j]);
dij = (int) ( RRR * acos( 0.5*((1.0+q1)*q2 - (1.0-q1)*q3) ) + 1.0);
```

کد جاوا برای محاسبه فاصله جغرافیایی و همچنین به عنوان نمونه فایل Excel محاسبه فواصل برای مساله ulysses16.tsp به همراه تمرین روی سایت گذاشته شد.

مراحل و خروجی های تمرین:

مرحله ۱: طراحی اجزای الگوریتم

۱-الف) اقدامات

در این مرحله اجزای الگوریتم ACS را طراحی کنید. شامل:

- نحوه نمایش جوابها
- محدوده مقادیر اولیه فرمون ها
- محدوده تعداد مورچه ها
- محدوده مقدار پارامتر q_0
- محدوده مقدار پارامتر β
- محدوده مقدار پارامتر α
- محدوده ضریب تبخیر محلی ζ
- محدوده ضریب تبخیر عمومی ρ

۱-ب) خروجی

اجزای تعیین شده در گزارش تمرین در یک فایل Word تایپ شود.

مرحله ۲: طراحی ساختمان داده

۲-الف) اقدامات

در این مرحله باید کلیه متغیرها و آرایه های مورد نیاز برای ذخیره سازی داده ها و پیاده سازی الگوریتم مشخص گردد. برای این مرحله می توانید به اسلایدهای قسمت مربوط به پیاده سازی الگوریتم ACO مراجعه نمائید.

۲-ب) خروجی

شی ها، متغیرها و آرایه های تعریف شده را در فایل گزارش اضافه نمائید. برای متغیرها ذکر نوع آنها، طول و همچنین شرح کاربرد آنها بصورت جدول ارائه شود.

مرحله ۳: طراحی الگوریتم

مشابه تمرین شماره ۱ انجام دهید.

مرحله ۴: پیاده سازی

مشابه تمرین شماره ۱ انجام دهید.

مرحله ۵: تنظیم پارامترها

پارامترهایی که در این مساله باید تنظیم شود عبارتند از:

- مقادیر اولیه فرمون ها
- تعداد مورچه ها
- مقدار پارامتر q_0
- مقدار پارامتر β
- مقدار پارامتر α
- مقدار ضریب تبخیر محلی ζ
- مقدار ضریب تبخیر عمومی ρ

اقدامات و خروجی این مرحله مشابه تمرین شماره ۱ انجام شود.

مرحله ۶: تحلیل عملکرد الگوریتم

مشابه تمرین شماره ۱ انجام دهید.

مرحله ۷: مستند سازی

مشابه تمرین شماره ۱ انجام دهید.