گزارش تمرین امتیازی

نام: محمد جمشیدی

شماره دانش‌جویی: ۹۸۱۰۰۷۱۸

در این تمرین به حل مسئله‌ی دوره‌گرد با استفاده از الگوریتم ژنتیک می‌پردازیم. کلیت این الگوریتم در زیر توضیح داده شده است.

۰. ابتدا ۱۵ عدد رندوم به عنوان مختصات افقی و ۱۵ عدد رندوم دیگر به عنوان متخصات عمودی شهرها تولید می‌کنیم. این‌کار با استفاده از تابع initialize_position_s() انجام می‌شود. سپس درون یک ماتریس ۱۵*۱۵، فاصله‌های متناظر میان دو شهر را محاسبه می‌کنیم و قرار می‌دهیم. این‌کار با کمک تابع determine_distance_s() انجام می‌شود.

۱. یک جمعیت اولیه می‌سازیم. هر عضو این جمعیت یک جایگشت از شهرها است. شهرها را با اعداد ۰ تا ۱٤ نشان می‌دهیم. این‌کار به کمک تابع initialize_generation() انجام می‌شود.

۲. برای تعیین احتمال انتخاب والدها، جمعیت را براساس کم‌ترین طول مسیر رتبه‌بندی می‌کنیم. این کار به کمک تابع sort_path_s() انجام می‌شود.

۳. به طور تصادفی یک جفت والد را با توجه به احتمال انتخابی که به هر عضو جامعه نسبت داده شده است انتخاب می‌کنیم. سپس بر روی این والدها عملیات cross_over را به کمک تابع cross_over() انجام می‌دهیم. برای پذیرفته شدن این عملیات یک احتمال در نظر می‌گیریم. مقدار این احتمال به صورت پیش‌فرض ۰٫۸ است.

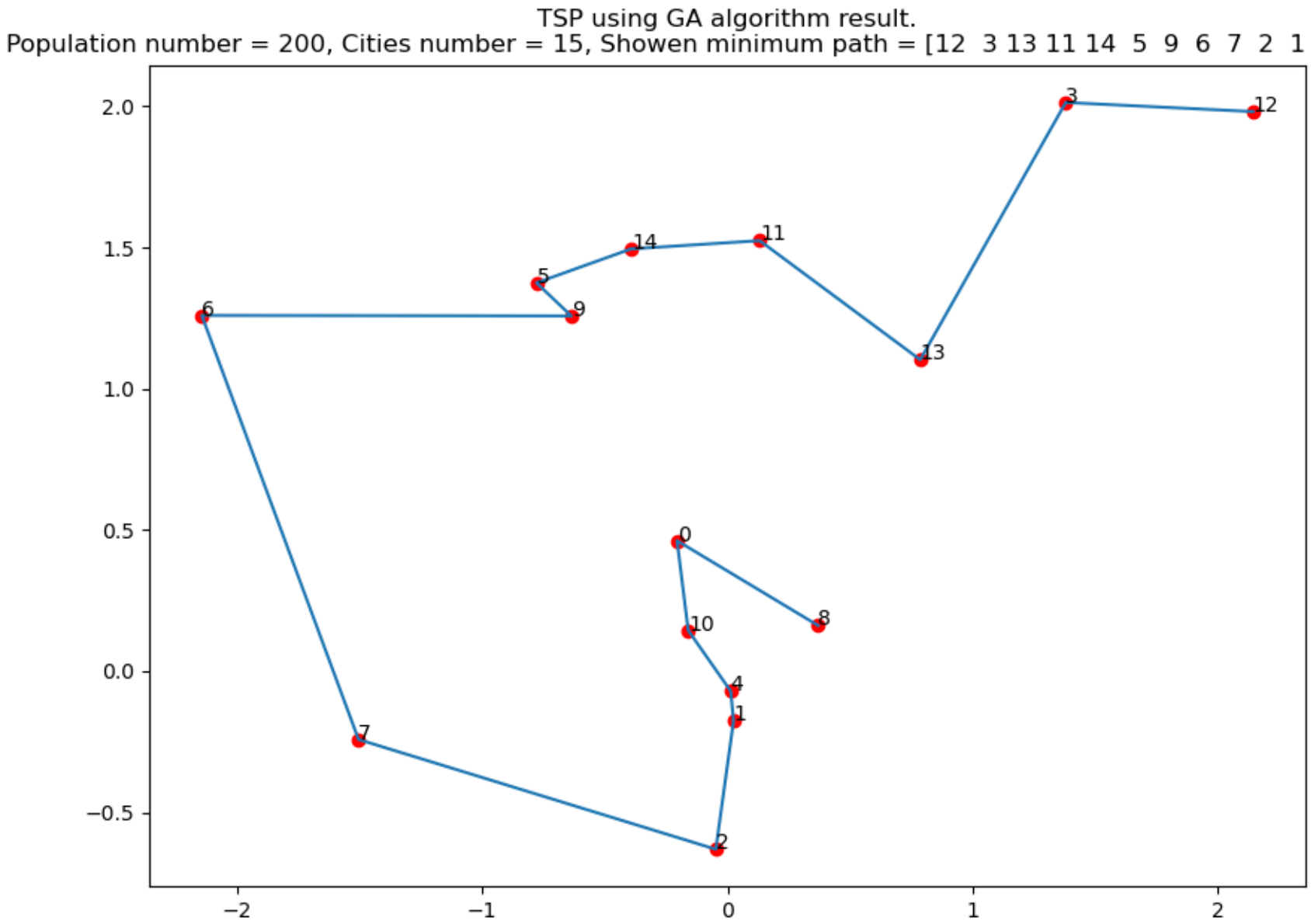
٤. بر روی هر کدام از دو عضو بدست آمده در بخش قبلی عملیات جهش را به کمک تابع mutate() انجام می‌دهیم. برای پذیرفته شدن این عملیات یک احتمال در نظر می‌گیریم. مقدار این احتمال به صورت پیش‌فرض ۰٫۰۸ است.

۵. به کمک تابع make_new_generation() گام‌های ۳ و ٤ را آن‌قدر تکرار می‌کنیم تا همه‌ی والدها شانس جفت‌گیری داشته باشند. و سپس نسل جدید را جایگزین نسل قدیمی می‌کنیم. سپس آن را مرتب می‌کنیم. هم‌چنین مقدار کمینه‌ی طول مسیر در این جامعه و مسیر متناظر با آن را برای خود ذخیره می‌کنیم.

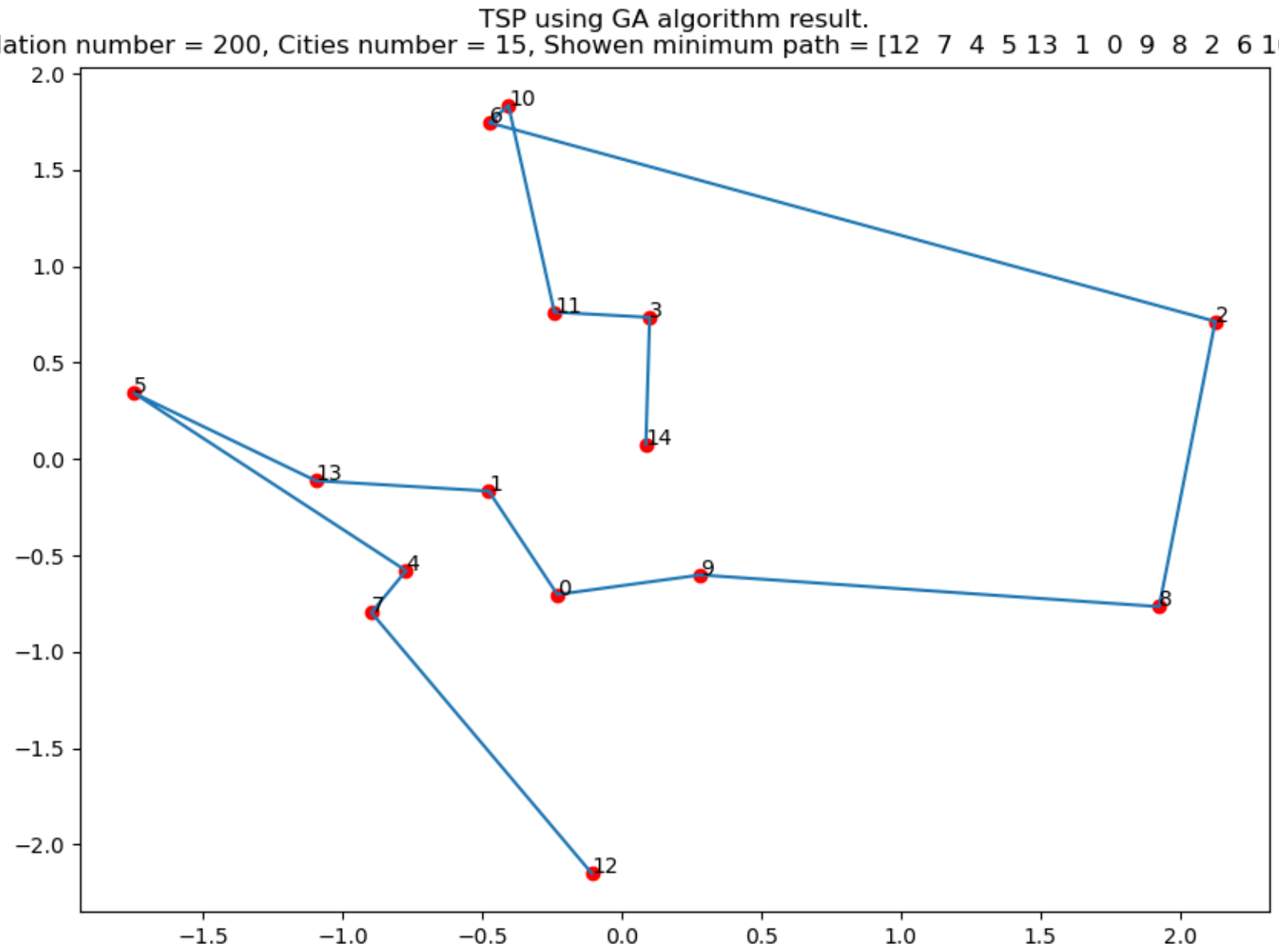
٦. قدم‌های ۳ تا ۵ را آنقدر تکرار می‌کنیم تا در لیست کمینه‌ی طول مسیرهای متناظر با نسل‌های مختلف، به یک ثبات برسیم و دیگر شاهد تغییر نباشیم. این‌کار با پارامتر threshold کنترل می‌شود. پیش‌فرض این پارامتر ۵ است یعنی نسل‌سازی تا جایی ادامه پیدا می‌کند که در لیست کمینه مسیرها، ۵ بار پشت سر هم یک مقدار عینا تکرار شود؛ البته این موضوع پس از اتمام هر ۲۰ بار تکامل اتفاق می‌افتد. در این صورت این مقدار ثابت را به عنوان کمینه‌ی طول مسیر، و یک مسیر متناظر با آن را به عنوان مسیر بهینه معرفی می‌کنیم (چون ممکن است که مثلا چند مسیر با طول کمینه وجود داشته باشند، برنامه فقط یکی از آن‌ها را معرفی می‌کند؛ البته اگر نیاز به دانستن همه‌ی این مسیرها باشد با یک تغییر ساده و کوچک در کد برنامه می‌توان این خواسته را برآورده کرد). این کارها با استفاده از تابع make_complete_evolution() انجام می‌شود.

همچنین نمودار نقاط مربوط به شهرها و مسیر بهینه هم رسم می‌شود.

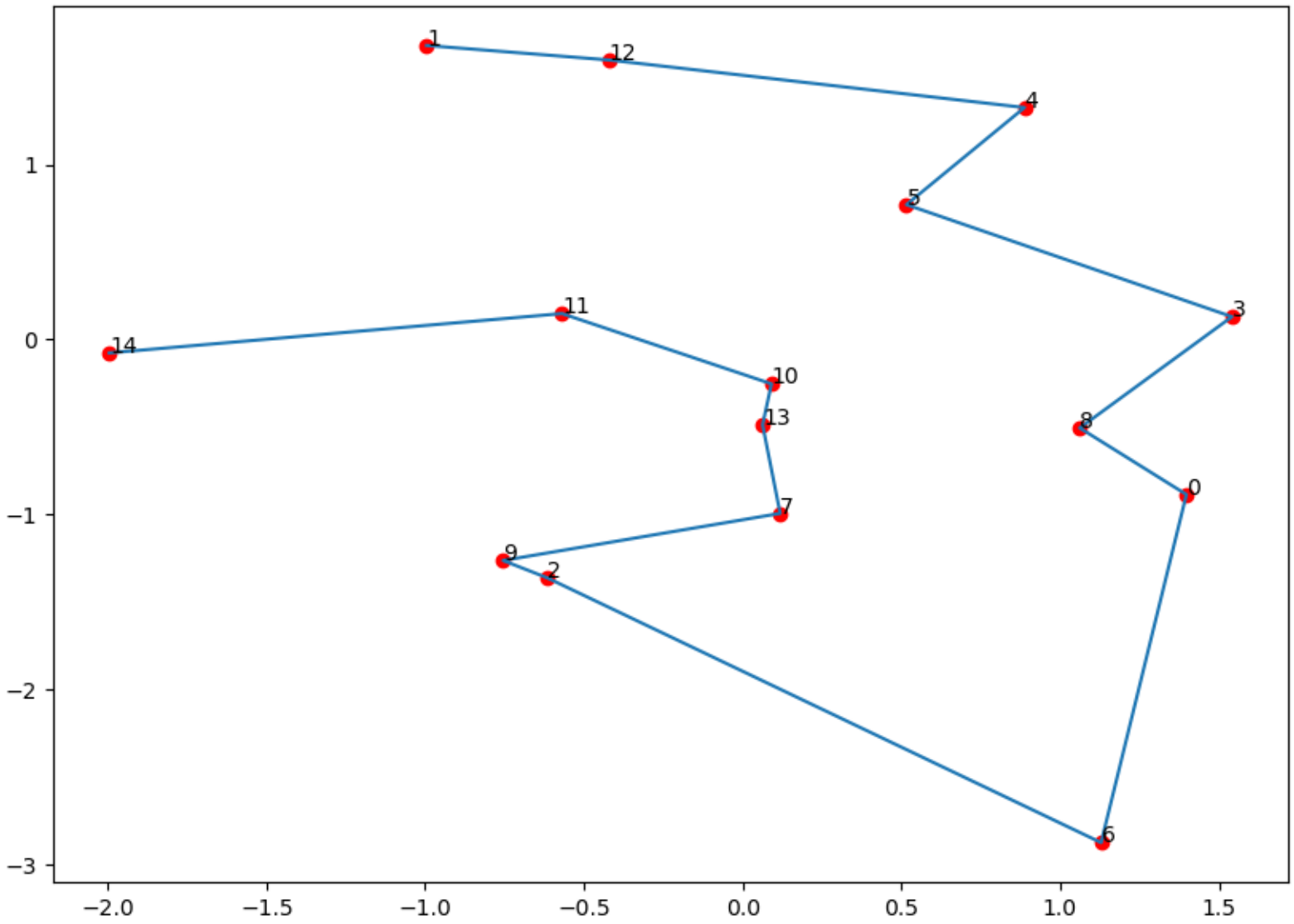
چند نمونه از اجراهای برنامه:



[17.23, 16.66, 16.19, 15.85, 15.07, 16.23, 13.33, 14.38, 15.305, 12.54, 13.734, 14.44, 13.28, 13.32, 13.25, 13.32, 13.07, 12.77, 13.05, 12.766, 12.62, 12.54, 12.73, 12.47, 12.65, 13.16, 12.766, 12.484, 12.625, 12.65, 11.92, 11.445, 11.805, 11.484, 11.74, 11.414, 11.19, 11.37, 10.766, 10.766, 10.766, 10.89, 10.766, 10.766, 10.734, 10.766, 10.36, 10.21, 10.24, 10.21, 10.21, 10.21, 10.21, 10.21, 10.21, 10.21, 10.21, 10.21, 10.21, 10.21, 10.21]
a path with minimum length: [12 3 13 11 14 5 9 6 7 2 1 4 10 0 8]



[19.67, 18.86, 18.25, 18.77, 19.1, 17.23, 16.88, 17.39, 16.88, 16.03, 16.11, 13.34, 12.71, 15.23, 12.96, 14.93, 15.54, 14.06, 14.91, 14.836, 13.53, 13.26, 14.32, 14.32, 14.33, 13.7, 14.516, 14.586, 13.17, 13.96, 15.01, 14.48, 14.63, 14.35, 13.79, 14.35, 13.13, 13.66, 14.125, 13.914, 13.43, 14.12, 13.14, 14.06, 14.23, 13.83, 13.83, 14.34, 13.805, 13.83, 14.17, 13.83, 13.83, 13.4, 13.22, 14.14, 13.78, 14.15, 14.12, 14.125, 14.24, 14.12, 13.85, 14.12, 13.4, 13.96, 14.12, 14.016, 14.12, 13.89, 13.96, 14.19, 13.945, 14.28, 13.92, 13.32, 14.055, 13.484, 13.92, 13.484, 13.484, 14.19, 13.71, 13.484, 13.414, 13.96, 13.75, 13.96, 13.95, 13.83]
a path with minimum length: [12 7 4 5 13 1 0 9 8 2 6 10 11 3 14]



[20.73, 18.36, 19.89, 18.98, 17.98, 17.95, 18.6, 18.17, 17.84, 17.08, 17.47, 17.03, 17.81, 17.75, 17.61, 17.55, 17.84, 17.61, 15.445, 16.44, 16.44, 16.19, 16.44, 15.445, 15.34, 15.445, 15.445, 15.34, 14.75, 14.75, 14.555, 14.555, 14.555, 14.555, 14.555, 14.555, 14.555, 14.555, 14.555, 14.42, 14.42, 14.42, 14.42, 14.42, 14.42, 14.42, 14.42, 14.42, 14.42, 13.88, 13.88, 13.88, 13.88, 13.88, 13.88, 13.88, 13.88, 13.88, 13.88, 13.88, 13.88, 13.46]

a path with minimum length: [14 11 10 13 7 9 2 6 0 8 3 5 4 12 1]