

## بسم الله الرحمن الرحيم

### گزارش مسئله‌ی فروشنده‌ی دوره‌گرد

زینب ایوبی ۹۷۱۰۰۶۴۳

فروشنده‌ی دوره‌گردی می‌خواهد برای فروش اجناس خود به ۱۵ شهر مختلف سفر کند. او می‌خواهد از یک شهر آغاز کرده و بدون بازگشت به شهرهایی که قبلاً رفته هر ۱۵ شهر را برود. می‌خواهیم ترتیبی از شهرها را به او پیشنهاد بدهیم که اگر طبق این ترتیب میان شهرهای مختلف سفر کند مسافت بهینه‌تری را طی کرده باشد.

در این شبیه‌سازی مختصات طول و عرض شهرها به صورت تصادفی عددی ۰ و ۱۰ کیلومتر انتخاب می‌شوند یعنی همه‌ی ۱۵ شهر در صفحه‌ای به مساحت ۱۰۰ کیلومتر مربع قرار دارند. (وظیفه‌ی تعیین مکان شهرها به عهده‌ی تابع `cities_coordinates(n)` است که در آن  $n$  تعداد شهرهاست.)

هم‌چنین شهرها از ۰ تا ۱۴ شماره‌گذاری شده‌اند بدین ترتیب که درایه‌ی صفرا<sup>۱</sup>م آرایه‌ی مکان شهرها، مختصات شهر ۰ام و درایه‌ی چهاردهم آرایه‌ی مکان شهرها، مختصات شهر ۱۴ام را نشان می‌دهد و مثلاً اگر برای ترتیب سفر کردن آرایه‌ای به شکل زیر در اختیار فروشنده قرار گیرد بدین معناست که سفر خود را از شهر ۱۵ام آغاز کرده، به شهر ۹ام برود و همین‌طور مطابق با الگو سفر خود را به شهر ۴ام ختم کند:

[5 , 9 , 0 , 11 , 7 , 12 , 2 , 14 , 1 , 13 , 3 , 6 , 8 , 10 , 4]

برای یافتن مسیر بهینه از الگوریتم ژنتیکی بهره می‌بریم:

بدین صورت که از میان همه‌ی ترتیب‌های ممکن سفر بین شهرها (که طبیعتاً تعداد آن‌ها  $15!$  است) نسلی با جمعیت  $m$  را به تصادف انتخاب می‌کنیم (این کار را تابع  $initial\_population(n, m)$  انجام می‌دهد). و سپس برای هر یک از افراد این نسل (هر یک از ترتیب‌های سفر بین این ۱۵ شهر) مسافت کل (مجموع مسافت بین هر دو شهر متوالی در آن ترتیب) را محاسبه می‌کنیم (این وظیفه به عهده‌ی تابع  $total\_distance(order\_of\_cities\_array, x\_cities, y\_cities, n)$  و  $fitness(population, m, x\_cities, y\_cities, n)$  است) و در نهایت این نسل را با توجه به مسافت کل هر فرد از این نسل  $sort$  می‌کنیم بدین صورت که آن ترتیبی از شهرها که بهینه‌ترین مسافت را دارد در صدر و آن که بیش‌ترین مسافت را می‌طلبد در ذیل این ترتیب قرار می‌گیرد. (خروجی تابع  $sort(population, fit, n, m)$ )

در این مرحله نسلی را در اختیار داریم که با توجه به بهینه‌بودن مسافت مورد نیاز برای پیمودن آن‌ها مرتب شده‌اند. اکنون می‌خواهیم نسل بعدی را با همین جمعیت و با لحاظ فرآیند نخبه‌پروری از روی این نسل بسازیم. بدین صورت که دو عضو اول نسل قبلی که به ترتیب بهینه‌ترین مسافت را دارند مستقیماً به عنوان دو عضو اول نسل جدید انتخاب می‌شود و سپس برای ایجاد سایر اعضا هر بار ۲ عضو از نسل قبلی را به تصادف و البته با احتمال‌های متفاوت انتخاب می‌کنیم و از روی آن‌ها دو عضو جدید تولید می‌کنیم. (نحوه‌ی ایجاد دو عضو جدید  $a, b$  از دو عضو قدیمی  $c, d$  به این صورت است که ۷ شهر اول  $a$  را همان ۷ شهر اول  $c$  می‌گذاریم و سپس ۸ شهر بعدی آن را از روی ترتیب شهرهای  $d$  با حذف اعضای تکراری می‌خوانیم و به همین طریق ۷ شهر اول  $b$  را همان ۷ شهر اول  $d$  می‌گذاریم و سپس ۸ شهر بعدی آن را از روی

ترتیب شهرهای C با حذف اعضای تکراری می‌خوانیم نکته‌ی مهم آن که در تولید هر فرد جدید اجازه می‌دهیم با احتمال کمی (احتمال ۸ درصد) جهش رخ دهد یعنی دو شهر از آن ترتیب، به تصادف انتخاب شوند و جای آن‌ها باهم عوض شود. و مطلب دیگر آن که همان‌طور که گفته‌شد برای رعایت فرآیند نخبه‌پروری والد‌ها با احتمال یکسان انتخاب نمی‌شوند و والد‌هایی که مسافت بهینه‌تری دارند یعنی در ردیف‌های بالاتر قرار می‌گیرند با احتمال بیش‌تری انتخاب می‌شوند و این سری احتمال را یک سری هندسی در نظر گرفته‌ام.) (تولید نسل جدید از نسل قدیم را تابع

`next_generation(sorted_population, n, m, possibilities)` انجام می‌دهد.)

حال نسل جدید به دست آمده را با توجه به بهینه‌بودن مسافت آن‌ها مرتب کرده و دوباره نسل بعدی را از روی آن می‌سازیم. این کار را به دفعات انجام می‌دهیم و هر بار بهینه‌ترین مسافت هر نسل را ذخیره می‌کنیم و در پایان نمودار تغییر مسافت بهینه را نسبت به تعداد تکرار نسل‌ها رسم می‌کنیم تا بهینه‌ترین مسافت ممکن را به دست آوریم.

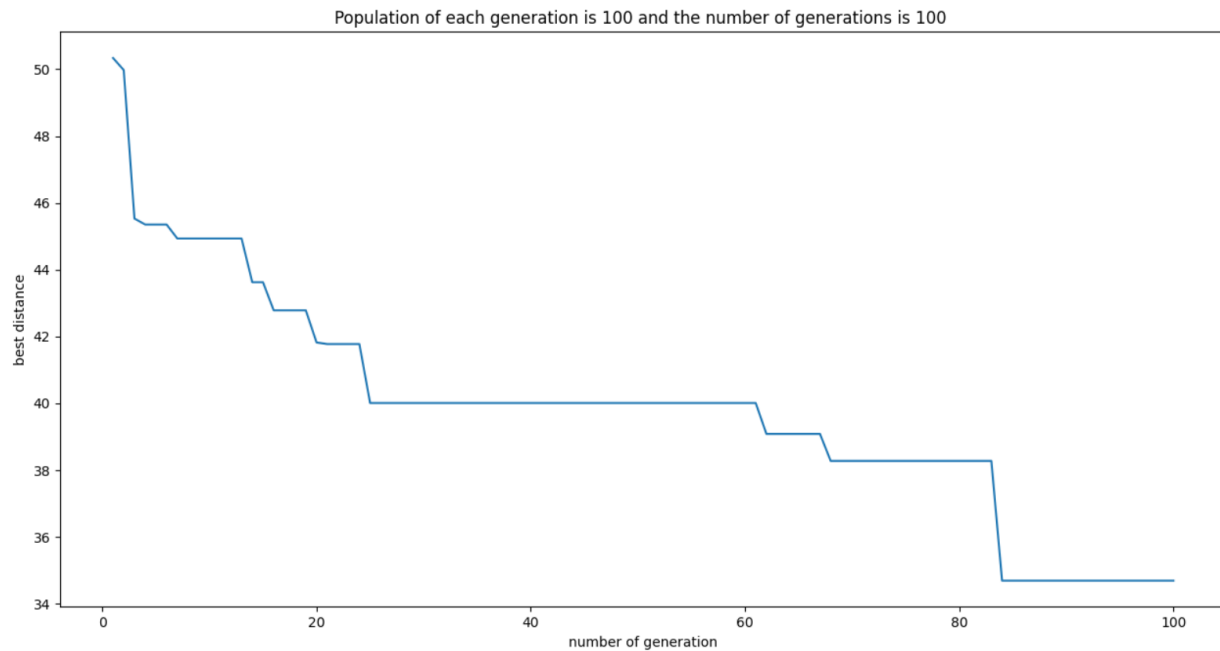
چند نمونه از نتایج را برای جمعیت‌های متفاوت نسل‌ها و تعداد جلوبردن نسل‌ها مشاهده می‌کنید:

(مختصات مکان‌های شهرها در هر بار اجرا در خروجی کد نمایش داده می‌شود. برای وجود معیاری (صدالبته نادقیق) از فاصله‌ی میان شهرها هر بار بهترین مسافت نسل اول که نسلی تصادفی است را گزارش می‌کنم. هم‌چنین در هر بار اجرا یکی از ترتیب‌هایی که بهترین مسافت را می‌دهد در خروجی چاپ می‌شود که چون آرایه‌ای از شماره‌ی شهرهاست و حاوی اطلاع معناداری برای این گزارش نبود در این گزارش نیاوردم.)

۱. جمعیت هر نسل ۱۰۰ فرد و پیش رفتن تا تولید ۱۰۰ نسل:

initial distance = 52.51 [km]

best distance = 34.69 [km]

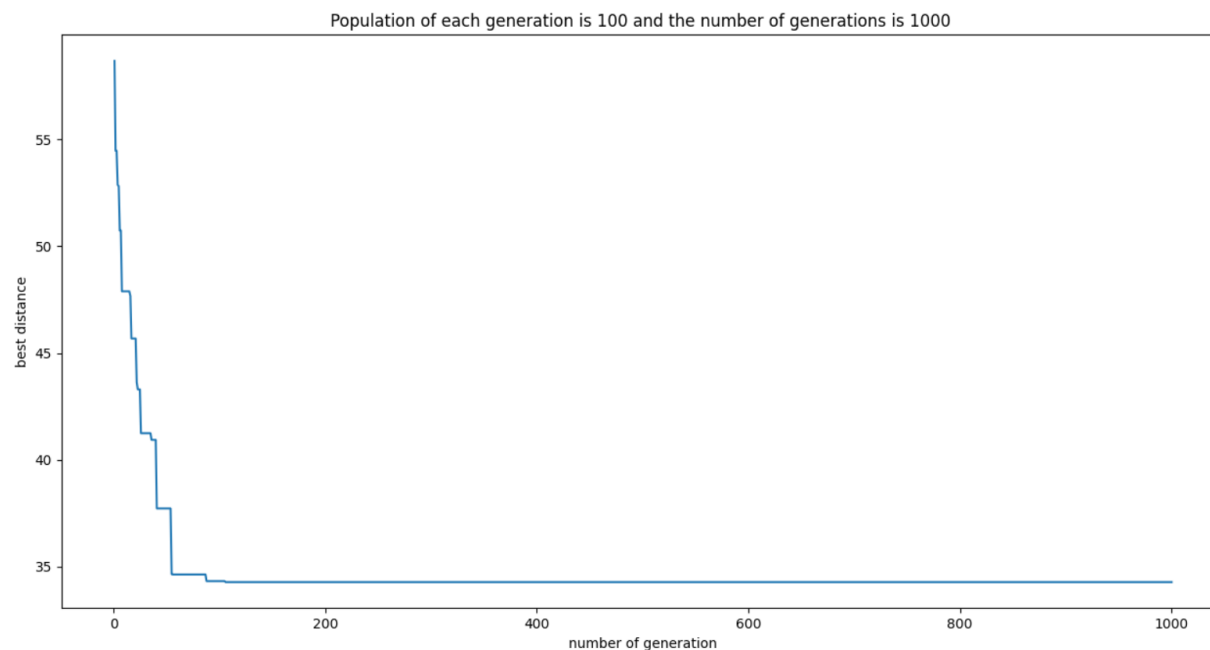


تصویر ۱: نمودار مسافت بهینه بر حسب تعداد نسل‌ها برای ۱۰۰ نسل و با جمعیت ۱۰۰  
فرد در هر نسل

۲. جمعیت هر نسل ۱۰۰ فرد و پیش‌رفتن تا تولید ۱۰۰۰ نسل:

initial distance = 60.92 [km]

best distance = 34.27 [km]

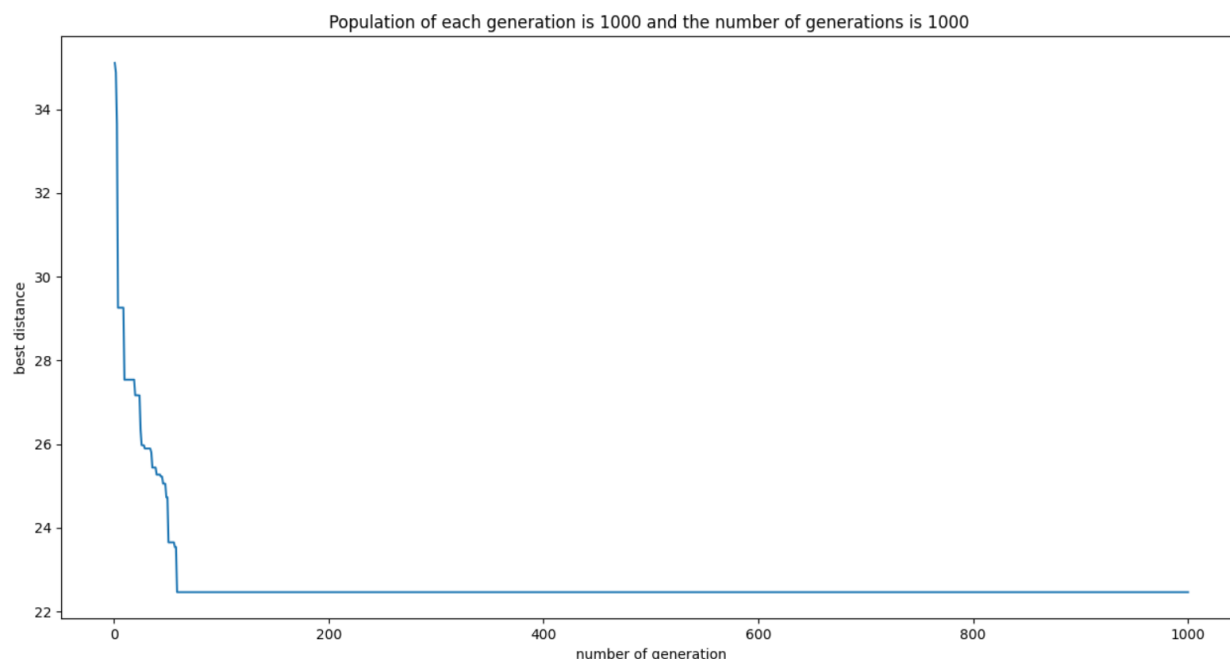


تصویر ۲: نمودار مسافت بهینه بر حسب تعداد نسل‌ها برای ۱۰۰۰ نسل و با جمعیت ۱۰۰ فرد در هر نسل

۳. جمعیت هر نسل ۱۰۰۰ فرد و پیش‌رفتن تا تولید ۱۰۰۰ نسل:

initial distance = 35.11 [km]

best distance = 22.46 [km]



تصویر ۳: نمودار مسافت بهینه بر حسب تعداد نسل‌ها برای ۱۰۰۰ نسل و با جمعیت ۱۰۰۰ فرد در هر نسل

همان‌طور که از تصاویر هویداست با تولید حدود ۱۰۰ نسل تقریباً به نتیجه‌ی قابل قبولی می‌رسیم.

سخن پایانی: فکرمی‌کنم آن چه از این مسئله مد نظر بود را در این گزارش آوردم. امیدوارم مقبول افتد :) بسیار مسئله‌ی جالبی بود لذت بردم.