



دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

درس هوش مصنوعی

تمرین دوم برنامه نویسی

اردیبهشت ۱۳۹۷

هدف:

هدف این پروژه آشنایی شما با الگوریتم‌های جستجوی غیر کلاسیک و نحوه فرموله کردن مسائل جستجوی مختلف در قالب یک رابط یکسان است.

شرح پروژه:

در این پروژه هر دانشجو می‌بایست مجموعه‌ای از الگوریتم‌های جستجویی که فهرست آنها در ادامه ذکر شده است را پیاده‌سازی و از این الگوریتم‌ها برای حل چند مساله جستجوی مختلف که در قالب یک واسط مشخص پیاده‌سازی می‌شوند استفاده کند.

پیاده‌سازی الگوریتم‌ها و مسائل مختلف می‌بایست به صورت مستقل از یکدیگر باشند به گونه‌ای که حل یک مساله جستجوی جدید نیازمند هیچ گونه تغییری در پیاده‌سازی الگوریتم‌های جستجو نباشند. هر الگوریتم جستجو می‌بایست در انتهای اجرا، اطلاعات کافی در مورد اجرای الگوریتم را بازگرداند.

الگوریتم‌ها:

الگوریتم‌هایی که می‌بایست در این پروژه پیاده‌سازی شوند عبارتند از:

- گرم و سرد کردن شبیه‌سازی شده (simulated annealing)

خروجی‌های الگوریتم:

– تعداد گره‌های ایجاد (مشاهده) شده

– تعداد گره‌های بسط داده شده

– راه حل و ارزش (شایستگی) راه حل یافته شده

- تپه نوردی (ساده، تصادفی، اولین انتخاب و شروع مجدد تصادفی)

خروجی‌های الگوریتم:

– تعداد گره‌های ایجاد (مشاهده) شده

– تعداد گره‌های بسط داده شده

– راه حل و ارزش (شایستگی) راه حل یافته شده

- الگوریتم ژنتیکی

خروجی‌های الگوریتم:

- بهترین، بدترین و متوسط شایستگی در هر نسل

- تعداد نسل‌ها تا رسیدن به جواب بهینه (در صورت رسیدن به جواب بهینه)

الگوریتم ژنتیکی را به گونه‌ای پیاده‌سازی کنید که اندازه جمعیت، تعداد کل ارزیابی‌های شایستگی (شرط خاتمه) و نرخ جهش آن قابل تنظیم باشد.

مسائل

به ازای هر یک از مسائلی که در این بخش معرفی می‌شود شما می‌بایست یک کلاس Problem مجزا بنویسید. کلاس مساله در این حالت می‌بایست شامل تابع حالت اولیه، تابع همسایگان هر حالت و تابع شایستگی (هدف) باشد. با اجرای الگوریتم‌های جستجوی ذکر شده برای هر مساله، عملکرد این الگوریتم‌ها در حل مساله مورد نظر را بررسی و با یکدیگر مقایسه کنید.

مساله اول: رنگ آمیزی گراف

برای مساله رنگ آمیزی گراف یک تابع هدف تعریف کنید و از الگوریتم‌های تپه‌نوردی (نسخه‌های مختلف الگوریتم تپه‌نوردی) برای حل این مساله استفاده کنید. خروجی‌های ذکر شده برای نسخه‌های مختلف الگوریتم تپه‌نوردی را گزارش و نتایج را با یکدیگر مقایسه نمایید.

ورودی برنامه یک گراف و یک عدد که بیانگر حداکثر تعداد رنگ مورد استفاده است، می‌باشد. گراف باید طوری رنگ شود که تعداد همسایه‌های هم‌رنگ حداقل بشود.

مساله دوم: جدول حروف

یک جدول (آرایه دو بعدی) از حروف و یک آرایه از کلمات (تحت عنوان دیکشنری) به مساله داده می‌شود. در جدول حروف می‌توانیم با شروع از یک حرف و حرکت در ۴ جهت یک ترتیب از حروف را بسازیم. شما باید در این جدول حروف را طوری جا به جا کنید که تعداد ترتیب‌هایی از حروف که می‌توان از این جدول بدست آورد و داخل آرایه دیکشنری نیز هست ماکزیمم شود. برای حل این مساله از الگوریتم SA استفاده کنید.

الف) سه روش مختلف برای کاهش دمای الگوریتم SA انتخاب کنید و نتایج را با یکدیگر مقایسه کنید. ب) خروجی‌های ذکر شده برای الگوریتم SA را گزارش و نتایج را با یکدیگر مقایسه نمایید.

برای نمونه جدول و دیکشنری زیر را در نظر بگیرید

a	p	t
m	l	b
k	l	o
u	o	c

{cool, cat, talk, go}

با تغییر چیدمان حروف به صورت زیر تعداد کلماتی از دیکشنری که میتوان تولید کرد ۳ عدد خواهد بود که مقدار ماکزیمم در این نمونه میباشد.

c	o	o
a	t	l
l	m	b
k	u	p

مساله سوم: کیبوردسازی!

میخواهیم ۲۶ حرف انگلیسی را بر روی کیبورد مدنظر در شکل زیر قرار دهیم به صورتی که کاربران بتوانند راحت تر از آن استفاده کنند. معیارهای راحتی استفاده از کیبورد بر اساس دو قاعده زیر بیان می شود :

الف) حروف انگلیسی که بیش از همه استفاده می شوند به طور متوازی بین دو بخش چپ و راست کیبورد تقسیم شوند.

حروف انگلیسی به ترتیب استفاده به صورت زیر هستند :

۱. E

۲. T

- A .۳
- I .۴
- N .۵
- O .۶
- S .۷
- H .۸
- R .۹

ب) دو حرف از حروف متوالی که در کلمات بیشتر دیده می‌شوند تقریباً به صورت متعادل در دو قسمت چپ و راست صفحه جای بگیرند. حروف متوالی متداول به صورت زیر هستند :

- th .۱
- er .۲
- on .۳
- an .۴
- re .۵
- he .۶
- in .۷
- ed .۸

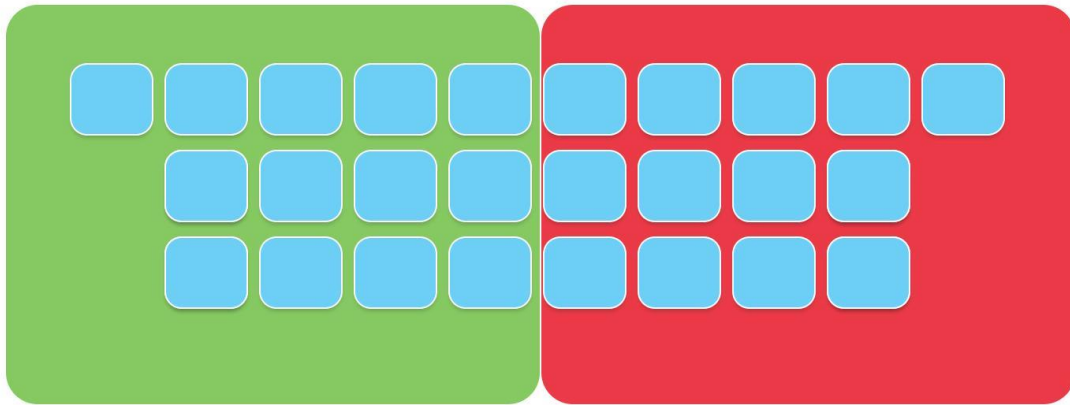
با توجه به موارد بالا تعریف تابع ارزیابی میتواند برعهده شما باشد و اولویت ترتیبها و مقداردهیها در اختیار خودتان است. اما موارد بالا در آن رعایت شده باشد. مسئله بالا را با الگوریتم ژنتیک حل کنید.

الف) نمودارهای بهترین، بدترین و میانگین شایستگی را در طول زمان ترسیم کنید.

ب) تاثیر کاهش یا افزایش احتمال جهش در نتایج را بررسی کنید.

ج) تاثیر کاهش یا افزایش تعداد برشها (n) در نتایج را بررسی کنید.

د) با ثابت نگه داشتن تعداد ارزیابیهای شایستگی، تاثیر اندازه جمعیت بر همگرایی الگوریتم و کیفیت نتایج را بررسی کنید.



مواردی که در انجام این پروژه برنامه نویسی باید رعایت کنید:

- برنامه‌های خود را به یکی از زبان‌های جاوا، ++C یا پایتون پیاده سازی کنید.
- پروژه‌ها به صورت انفرادی تعریف شده است و در صورت تشابه جزئی یا کلی کدهای دو دانشجو، به هر دو نفر نمره صفر تعلق خواهد گرفت.
- برای هر تمرین علاوه بر کد پیاده سازی شده، گزارشی تهیه کنید که در آن نتایج به دست آمده در هر آزمایش ارائه و مقایسه شود.
- در صورتی که گزارش درخواست شده در فایل های ارسالی نباشد، نمره ای تعلق نمی گیرد.
- کدها و مستندات خود را در یک فایل فشرده شده با قالب زیر نامگذاری و در سایت درس بارگذاری نمایید.

Project2_Student#_Lastname.zip

Project2_9431057_Heydari.zip

- در صورت وجود هرگونه سوال و یا مشکل می توانید با ایمیل nmoradzadehf@gmail.com در تماس باشید.

موفق باشید