



پروژه شماره ۴

پروژه‌ی الگوریتم ژنتیک

یک الگوریتم ژنتیک برای حل مسئله‌ی کوله‌پشتی صفر-یک

A GENETIC ALGORITHM FOR SOLVING 0/1 KNAPSACK PROBLEM

۱ تعریف مسئله

مسئله‌ی کوله‌پشتی صفر-یک (0/1-Knapsack) به صورت زیر تعریف می‌شود:

مجموعه‌ای از N شیء با وزن‌های مختلف w_1, w_2, \dots, w_N و ارزش‌های مختلف p_1, p_2, \dots, p_N با یک کوله‌پشتی با حداکثر گنجایش وزنی W موجود است. می‌خواهیم زیرمجموعه‌ای از این اشیا با ماکزیمم مجموع ارزش ممکن را در این کوله‌پشتی قرار دهیم. کدام اشیا باید انتخاب شوند؟

همان طور که می‌دانید این مسئله یک مسئله NP-hard است و تاکنون الگوریتمی بهتر از زمان-نمایی برای آن پیدا نشده است. در این پروژه می‌خواهیم با استفاده از الگوریتم ژنتیک راه حل بهینه (یا نزدیک به بهینه) را در مدت زمان نسبتاً کوتاهی بیابیم.

۲ راه‌حل با الگوریتم ژنتیک

با استفاده از MATLAB و توابع موجود در جعبه ابزار Genetic Algorithm and Direct Search، برنامه‌ای بنویسید که مسئله‌ی کوله‌پشتی صفر-یک را برای نمونه‌ی داده شده حل کند. برنامه‌ی شما باید ورودی‌های زیر را دریافت کند:

• ورودی‌های مسئله:

- لیست وزن اشیا (در قالب یک بردار N تایی $w = [w_1 \ w_2 \ \dots \ w_N]$)
- لیست ارزش اشیا (در قالب یک بردار N تایی $p = [p_1 \ p_2 \ \dots \ p_N]$)
- گنجایش وزنی کوله‌پشتی (در قالب یک عدد W)

• پارامترهای الگوریتم ژنتیک:

- اندازه‌ی جمعیت (در قالب یک عدد $popSize$)
- نرخ جهش (در قالب یک عدد بین صفر و یک $mutRate$)
- تعداد تکرار (تعداد نسل‌ها) (در قالب یک عدد $iterNum$)

۱-۲ بازنمایی کروموزوم

کروموزوم (ژنوم) را به صورت ساده در قالب یک بردار N تایی از صفرها و یک‌ها بازنمایی می‌کنیم:

$$(x_1, x_2, \dots, x_N)$$

که در آن $x_i = 0$ است اگر شیء i ام انتخاب نشود و $x_i = 1$ است اگر شیء i ام انتخاب شود ($1 \leq i \leq N$).

۲-۲ تابع برازش

تابع برازش را به صورت زیر تعریف می‌کنیم:

$$f(x_1, x_2, \dots, x_N) = \begin{cases} \sum_{i=1}^N p_i x_i & , \sum_{i=1}^N w_i x_i \leq W \\ 0 & , \sum_{i=1}^N w_i x_i > W \end{cases}$$

۳-۲ عملگرهای ژنتیکی

از عملگرهای تقاطع تک‌نقطه‌ای و جهش استاندارد استفاده می‌کنیم.

۴-۲ اجرای الگوریتم

الگوریتم را برای مسئله‌ی نمونه در فایل پیوست `testdata.m`، با پارامترهای زیر اجرا کنید و نتایج هر یک را جداگانه ثبت کنید.

- اندازه جمعیت ۱۰، ۱۰۰، ۱۰۰۰
- نرخ جهش ۰٫۱، ۰٫۰۱، ۰٫۰۰۱
- تعداد نسل ۵۰، ۱۰۰، ۵۰۰

در واقع ۲۷ ترکیب مختلف برای اجرای الگوریتم ژنتیک باید استفاده شود و نتایج آنها ثبت شده و با هم مقایسه شود. برای هر مورد بهترین پاسخ یافت‌شده، مقدار برازش آن و زمان رسیدن به پاسخ ثبت شود.

۳ گزارش پروژه

گزارش پروژه باید دارای ساختار زیر باشد:

- عنوان پروژه
- نام، و نام خانوادگی، شماره‌ی دانشجویی و ایمیل اعضای گروه
- مقدمه (تعریف پروژه به طور مختصر)
- پارامترهای الگوریتم ژنتیک استفاده شده و توابع اصلی آن
- جدول مقایسه‌ی ترکیب‌های مختلف اجرا و نتایج هر یک به صورت:

شماره	اندازه جمعیت	نرخ جهش	تعداد نسل	بrazش بهترین فرد	زمان اجرا
۱	۱۰	۰٫۱	۵۰

- نمودار «بهترین برازش» و «بrazش متوسط» بر حسب تعداد نسل برای هر ترکیب (هر دو منحنی برای هر ترکیب در یک نمودار رسم شود).
- نتیجه‌گیری

۴ تحویل پروژه

تحویل پروژه از طریق آپلود در سایت می‌باشد. فایل‌های زیر را در یک فایل `zip` با نام `knap-ga` قرار دهید و در محل مشخص شده در سایت درس آپلود کنید:

فایل برنامه‌ی MATLAB	<code>knapga.m</code>
فایل داده‌های برنامه‌ی MATLAB	<code>knapga.mat</code>
فایل گزارش پروژه	<code>knapga.pdf</code>

knap-ga.zip

* از آنجا که پروژه در قالب گروه‌های دونفری انجام می‌شود، فقط سرگروه باید فایل‌ها را آپلود کند.

برای استفاده از خط فرمان MATLAB برای الگوریتم‌های ژنتیک می‌توانید از پیوست F مرجع [1] استفاده کنید.

مراجع

- [1] N. Siddique, H. Adeli, **Computational Intelligence: Synergies of Fuzzy Logic, Neural Networks and Evolutionary Computing**, John Wiley & Sons, 2013.