# به نام خدا

# گزارش تمرین دوم درس شناسایی الگو

# طراحی الگوریتم‌های تکاملی و تنظیم پارامتر، برای برخی نمونه مسائل محک

#### استاد درس:

### جناب دکتر کارشناس

#### نام و نام خانوادگی دانشجو:

### امیررضا صدیقین

#### شماره دانشجویی:

## 993614024



# چکیده

در این گزارش، سعی شده است الگوریتمی برای حل مسائل محک به نام‌های OneMax ،Peak وTrap، طراحی شود. در روش ارائه شده در این گزارش سعی شده است که هم به اکتشاف در فضای راه‌حل‌ها و هم به بهره‌برداری توجه زیادی شود و متناسب با شرایط هر نسل، میزان توجه به هر یک از این دو بخش در طول اجرای الگوریتم تغییر کند. در این گزارش به روند الگوریتم و همچنین نتایج آزمایش‌ها پرداخته شده است.

# 1-مقدمه

دو نیروی اصلی در الگوریتم‌های تکاملی، نیروی تغییر و نیروی انتخاب است. هدف اصلی نیروی تغییر جهش، اکتشاف در فضای راه‌حل‌ها و جلوگیری از همگرایی زودرس است. همچین هدف اصلی نیروی‌ انتخاب، بهره‌برداری بیشتر و حرکت رو به جلو برای رسیدن به نقاط بهینه است.

اعمال فشار زیاد منجر به پیدایش راه‌حل‌های خوب ولی باعث کاهش تنوع جمعیت و درگیری با نقاط بهینه‌ی محلی می‌شود و ما را از رسیدن به نقطه‌ی بهینه‌ی سراسری باز می‌دارد. پس اگر یک فرایند تنظیم میزان فشار ارائه شود به گونه‌ای که اگر تنوع جمعیت کم شد تمرکز الگوریتم به سمت اکتشاف باشد و در غیر این صورت به دنبال بهره برداری باشد‌‌، می‌توان امید دست‌یابی به نقاط بهینه‌ی سراسری را داشت.

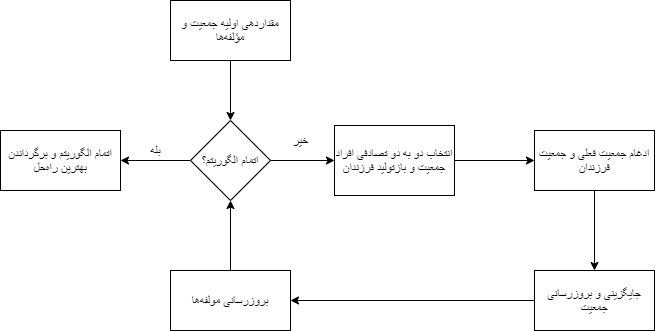
# 2- الگوریتم ارائه شده

## 2-1- مؤلفه‌های الگوریتم

مولفه‌های اصلی موجود در این الگوریتم، نمایش دودویی برای نمایش راه حل‌ها، توابع محک OneMax، Peak، Trap هر کدام به صورت جداگانه برای توابع برازندگی، است. برای انتخاب والدین، به خاطر دخیل بودن نقاط بیشتر برای بازتولید، از همه‌ی افراد جمعیت استفاده می‌شود و به طور تصادفی جفت جفت می‌شوند و اقدام به بازتولید می‌کنند. عملیات تقطیع یکنواخت با احتمال 80درصد، به عنوان عملیات بازترکیب انتخاب شده است. دلیل این انتخاب آن است که در این مسائل محک، وابستگی محلی وابستگی بین ژن‌ها کمتر دیده می‌شود و این نوع تقطیع مستقل از حفظ روابط بین ژن‌ها، دست به اکتشاف بیشتری می‌زند. هر ژن از فرزندان تولید شده ، با یک احتمال جهش، عوض می‌شوند. این احتمال جهش متناسب با رویکرد مسئله در هر نسل عوض می‌شود که در ادامه بیشتر به آن پرداخته خواهد شد. برای جایگزینی نیز، جمعیت فعلی با جمعیت فرزندان مخلوط شده و بر اساس برازندگی هر فرد، جمعیت حاصل به دو نیمه‌ی افراد با برازندگی بالا و افراد با برازندگی پایین تقسیم می‌شود. به دلیل نخبه سالاری، بهترین فرد انتخاب می‌شود و داخل جمعیت جدید قرار می‌گیرد. با استفاده از یک معیار به اسم نرخ فشار انتخاب، به اندازه این مقدار درصد از نیمه‌ی بالا و مابقی را از کل جمعیت حاصل شده (چه نیمه‌ی بالا، چه نیمه‌ی پایین) انتخاب می‌شود. این نرخ متناسب با رویکرد مسئله در هر نسل عوض می‌شود. شرط اتمام نسل‌ها رسیدن به جواب بهینه (یعنی رشته‌ی تمام یک) یا رسیدن به یک تعداد حداکثر نسل است.

## 2-2-روند الگوریتم

روند الگوریتم به صورت فلوچارتی در شکل1 آمده است.



شکل1: روند کلی الگوریتم پیشنهاد شده

### 1-2-2-مقداردهی‌های اولیه

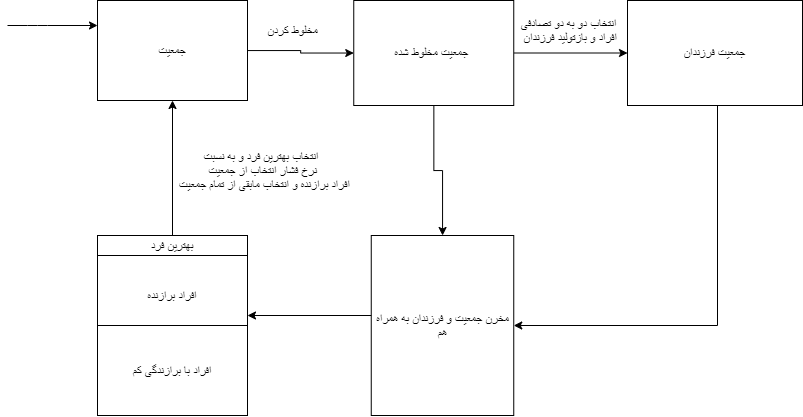
در این الگوریتم ابتدا جمعیت مقدار دهی اولیه با مقادیر تصادفی می‌شود و نرخ فشار انتخاب برابر 0.8 و احتمال جهش برابر 0.2 فرض می‌شود.

### 2-2-2-روند انتخاب افراد و بازتولید

در هر نسل، همه‌ی افراد داخل جمعیت برای بازتولید انتخاب می‌شوند و به صورت دو به دو تصادفی با احتمال 80 درصد اقدام به بازتولید می‌کنند. دو فرزند از هر دو والد با استفاده از تقطیع یکنواخت به وجود می‌آیند و در حین عملیات بازتولید، هر ژن از هر فرزند با احتمال در نظر گرفته شده برای آن، جهش پیدا می‌کنند.( هر ژن مستقل از بقیه‌ی ژن‌ها با احتمال مربوطه، مقدار آن عوض می‌‌شود.). این روند در شکل2 به تصویر کشیده شده است.

### 3-2-2-جایگزینی افراد جمعیت

پس از تولید جمعیت فرزندان، مخزنی از جمعیت فعلی به همراه جمعیت فرزندان تولید می‌شود و براساس برازندگی مرتب سازی می‌شوند. اولین فرد انتخاب شده برای نسل بعد، فرد با بهترین برازندگی است.(نخبه ترین فرد انتخاب می‌شود.). به نسبت نرخ فشار انتخاب، افراد از نیمه‌ی با برازندگی بیشتر انتخاب می‌شوند. برای مثال، اگر در نسل فعلی نرخ فشار انتخاب برابر با 0.7 باشد آن گاه 70 درصد از جمعیت جدید ما از نیمه‌ی اول مخزن که دارای برازندگی بیشتری هستند انتخاب می‌شود (این انتخاب به صورت بدون تکرار است.) و در انتها مابقی افراد از کل مخزن (چه نیمه‌ی بالای مخزن، چه نیمه‌ی پایین مخزن و حتی بدون در نظر گرفتن این که آن فرد قبلا انتخاب شده است) انتخاب خواهند شد.



شکل2: روند انتخاب افراد و بازتولید آن‌ها در الگوریتم پیشنهاد شده

### 4-2-2-بروزرسانی مؤلفه‌ها

در قدم بعدی باید ضریب انتخاب، نرخ فشار انتخاب و احتمال جهش را به روز رسانی کنیم. برای این امر، از رابطه‌ی 1 برای بروزرسانی ضریب انتخاب استفاده شده است. که در آن ضریب انتخاب به روز شده است و ضریب انتخاب در نسل گذشته است(در ابتدا این مقدار برابر با 30 می‌باشد). نیز نرخ فشار انتخاب در نسل گذشته است. دلیل انتخاب این رابطه آن است که در روند تکامل این مؤلفه تطبیق پیدا کند و متناسب با نرخ فشار انتخاب گذشته آن را تعدیل کند.( اگر نرخ فشار انتخاب در نسل گذشته بیشتر از 0.5 بوده، آن را کمتر می‌کند تا به اکتشاف بییشتر پرداخته شود و اگر این مقدار کمتر از 0.5 بوده،آن را بیشتر می‌کند تا الگوریتم به سمت بهره برداری بیشتر حرکت کند.

رابطه‌ی (1)

رابطه‌ی(1)

همچنین از رابطه‌ی 2 برای به روزرسانی نرخ فشار انتخاب استفاده شده است. که در آن نرخ فشار انتخاب جدید و میانگین برازندگی جمعیت تولید شده و میانگین برازندگی جمعیت نسل گذشته است و ProblemSize اندازه‌ی مسئله و ضریب انتخاب به دست آمده از رابطه‌ی 1 است. همچنین نرخ فشار انتخاب نباید از 0.3 کمتر و از 0.9 بیشتر باشد و اگر *این مقدار کمتر از 0.3 شد، آن را به 0.3 و اگر بیشتر از 0.9 شد، آن را به 0.9 تبدیل می‌کنیم*

رابطه‌ی (2)

همچنین از رابطه‌ی 3 برای بروزرسانی احتمال جهش استفاده شده است. که در آن احتمال جهش بروز شده، نرخ فشار انتخاب به دست آمده از رابطه‌ی 2 است.

رابطه‌ی (3)

هدف از این بروزرسانی‌ها آن است که اگر جمعیت به راه‌حلی همگرا شد رویکرد آن به سمت اکتشاف رود و اگر میزان تغییرات برازندگی زیاد بود (چه منفی چه مثبت) به سمت بهره‌وری قدم بردارد و همچنین یک تعادلی بین این دو هدف ایجاد کند.

### 5-2-2-شرط اتمام الگوریتم

روند الگوریتم به صورت تکراری می‌باشد و مدام چرخه‌ی آن تکرار می‌شود. شرط خروج این الگوریتم آن است که به جواب بهینه (رشته‌ی تمام یک) برسد یا آن که تعداد نسل تولید شده به یک مقدار بیشینه‌ی نسل‌ها که مشخص شده است، برسد.

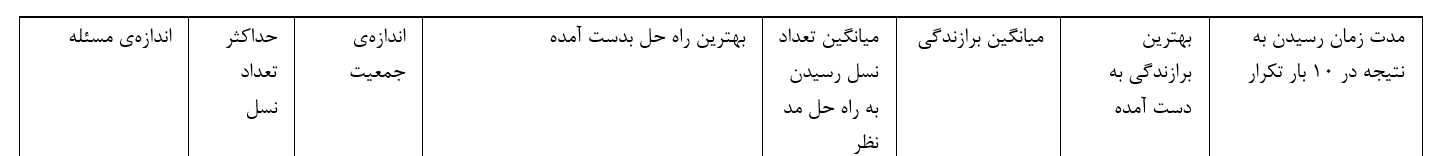
# 3-آزمایش

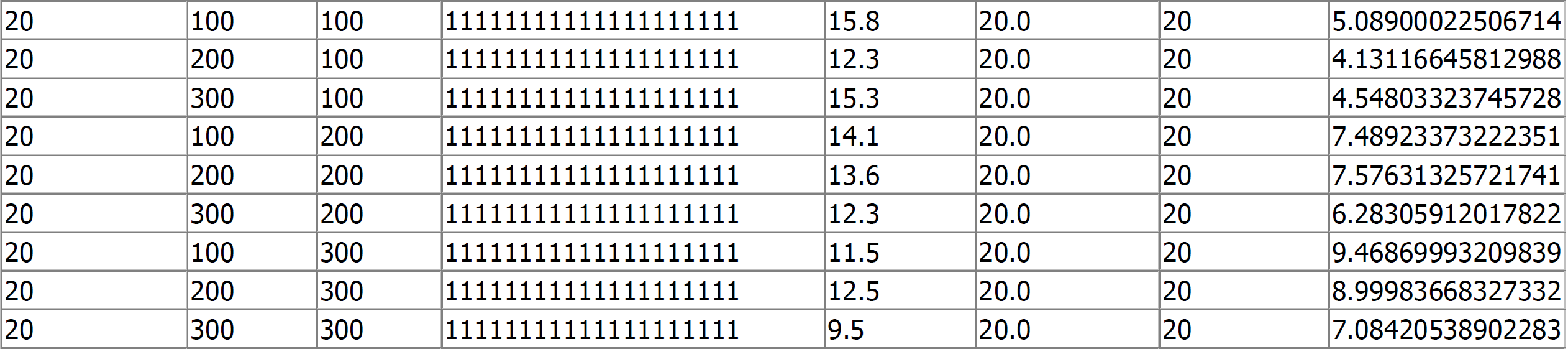
## 1-3-روند آزمایش

این الگوریتم برای سه تابع ارزیابی OneMax ،Peak وTrap و برای هر مسئله برای اندازه‌ی مسئله‌های 10 ،20 ،40 ،60، 80 با حد بیشینه‌ی تعداد نسل 100 ،200، 300 و اندازه‌ی جمعیت 100 ،200، 300 آزمایش شده است. به عبارت دیگر برای هر مسئله 45 حالت بررسی شده است و به دلیل تصادفی بودن الگوریتم، برای هر حالت الگوریتم 10 بار تکرار شده است.

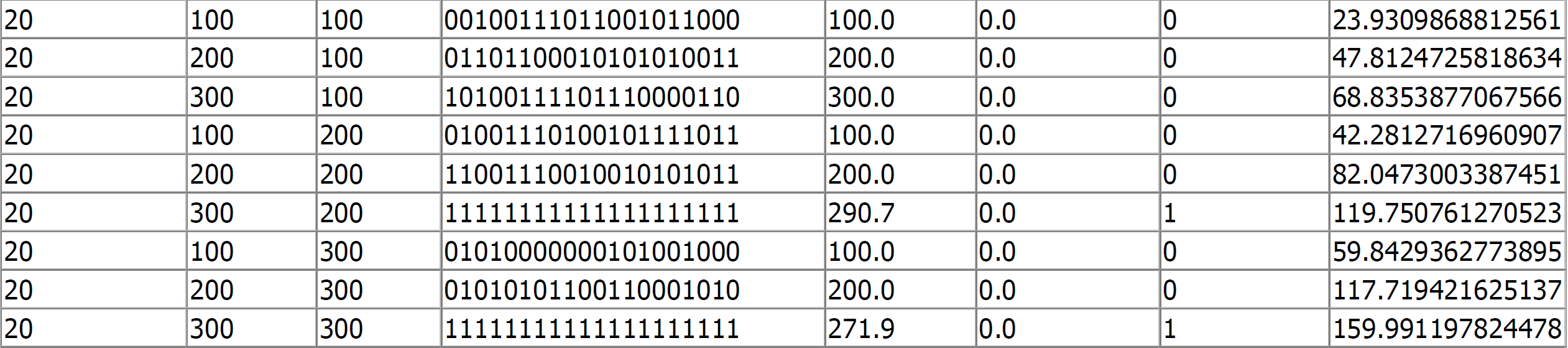
## 2-3-نتایج آزمایش

برای همه‌ی حالات آزمایش نتایج به درست آمده است و برای مثال برای اندازه مسئله 40 برای هر یک از مسائل صورت جدول1، 2 و3 درآمده است.





جدول 1: نتایج به دست آمده برای مسئله‌ی one\_max برای اندازه‌ی 40

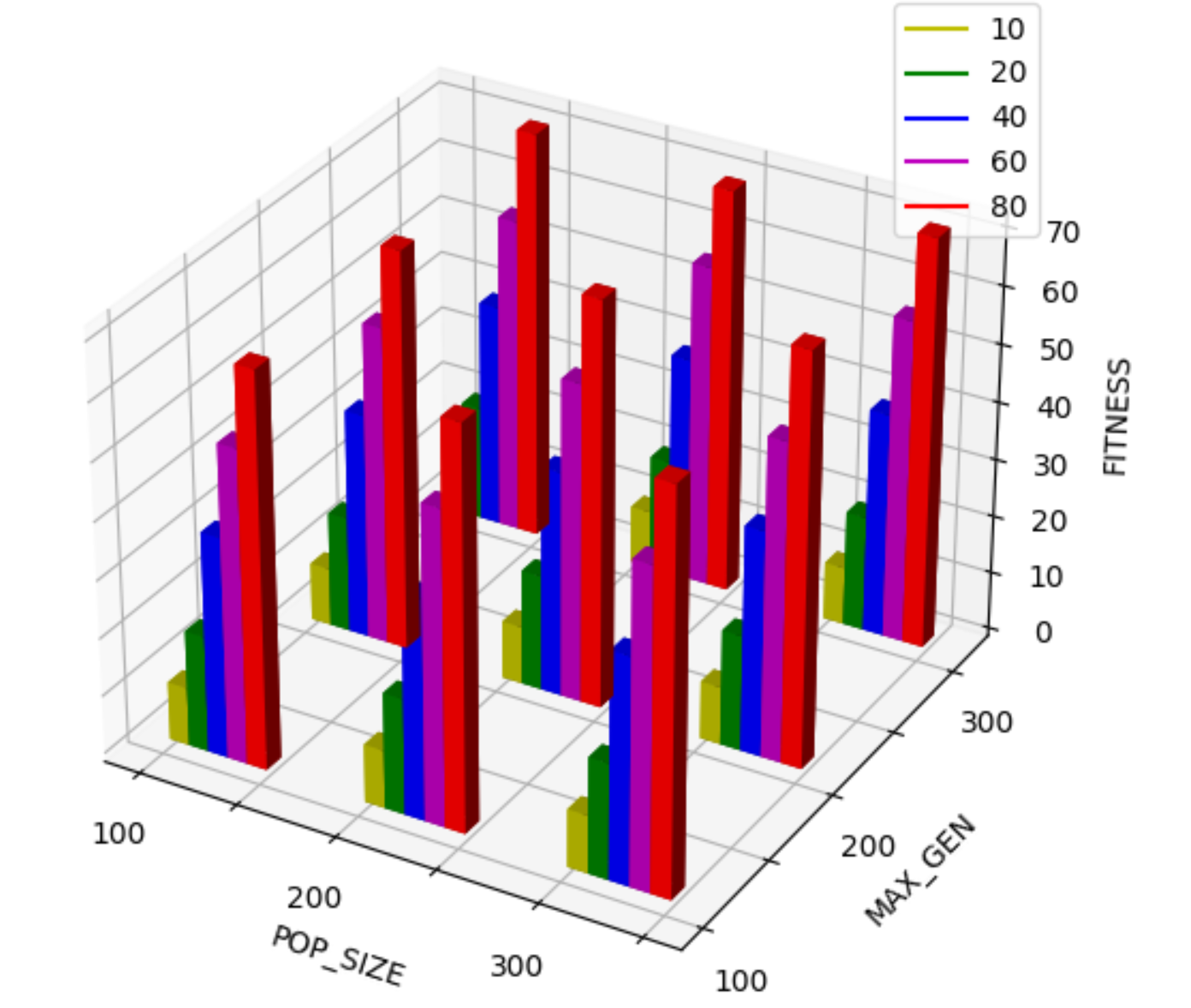


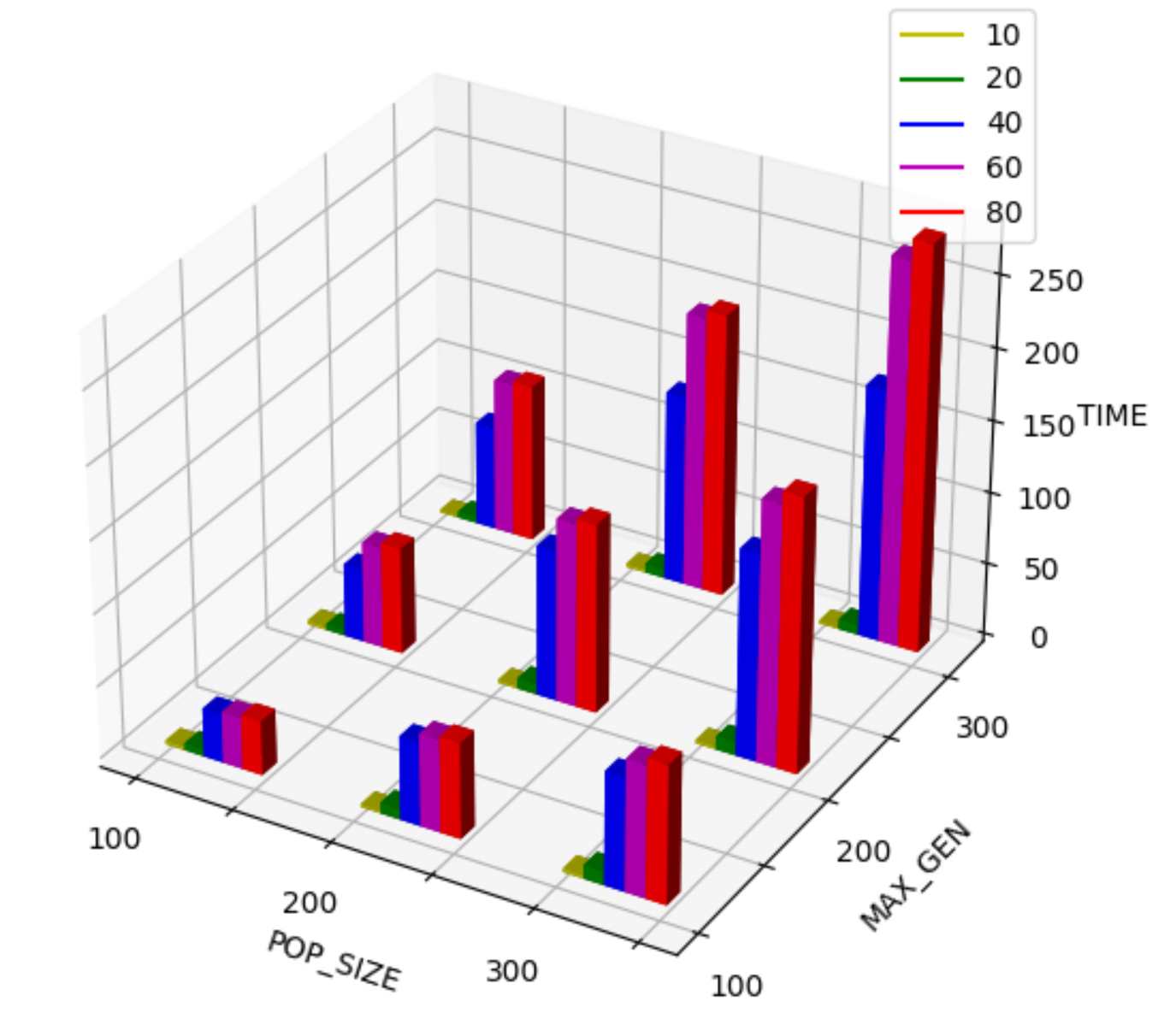
جدول 2: نتایج به دست آمده برای مسئله‌ی peak برای اندازه‌ی 40

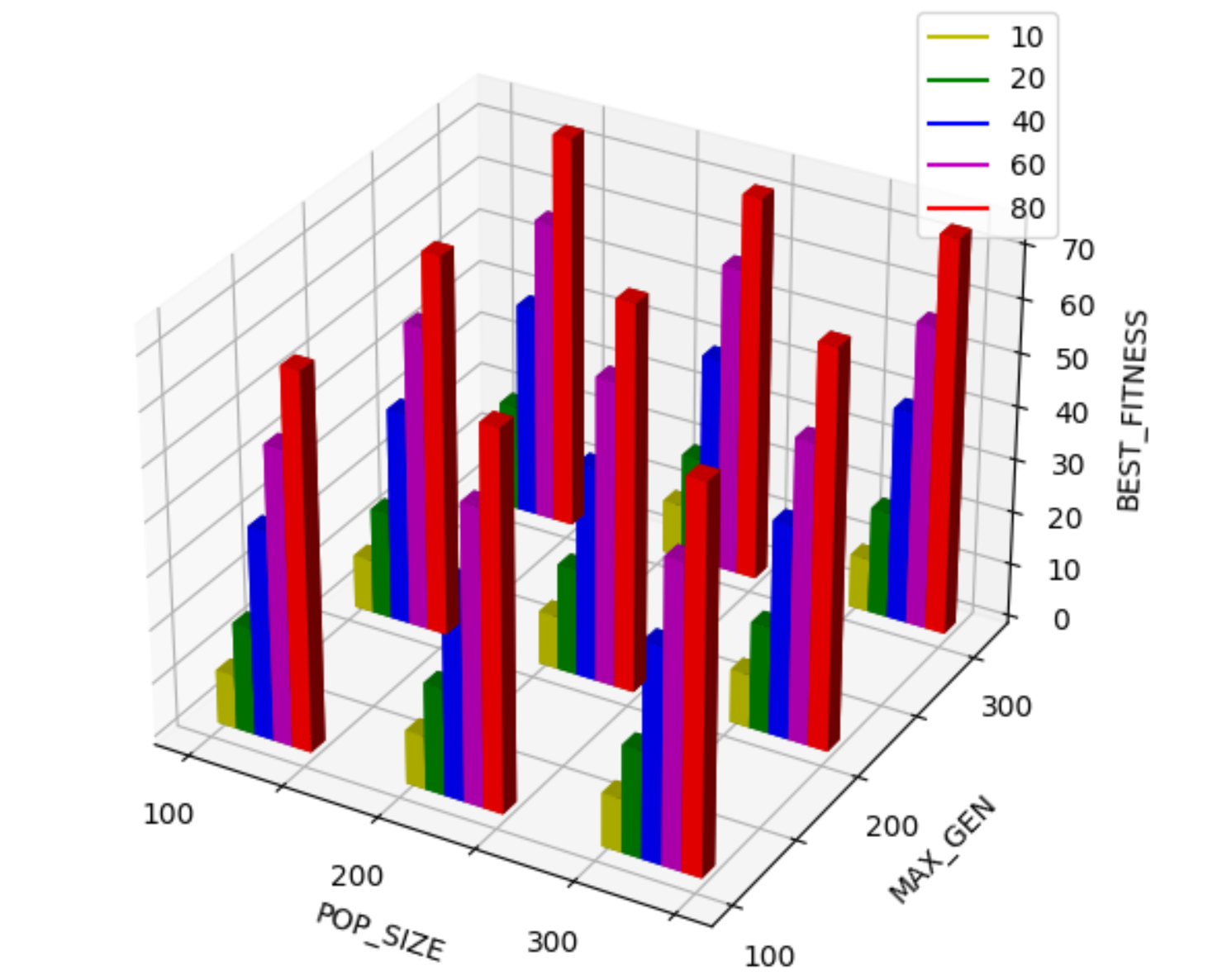


جدول 3: نتایج به دست آمده برای مسئله‌ی trap برای اندازه‌ی 40

همچنین نمودار میانگین برازندگی و بهترین برازندگی و زمان مصرفی بر اساس طول مسئله، اندازه‌ی جمعیت و حداکثر تعداد نسل برای مسئله‌ی OneMax به ترتیب در شکل‌های 3، 4، 5 آمده است.





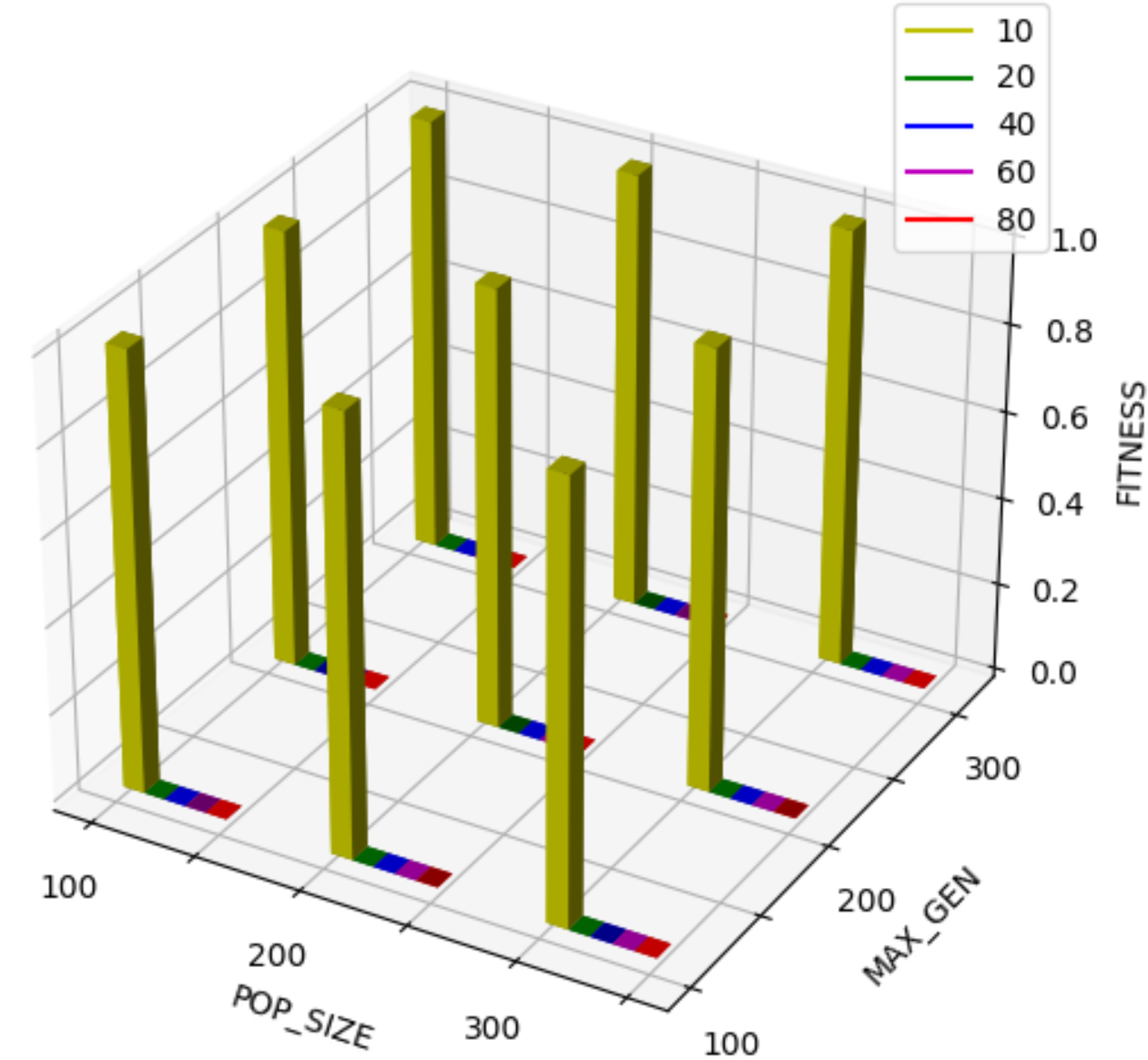


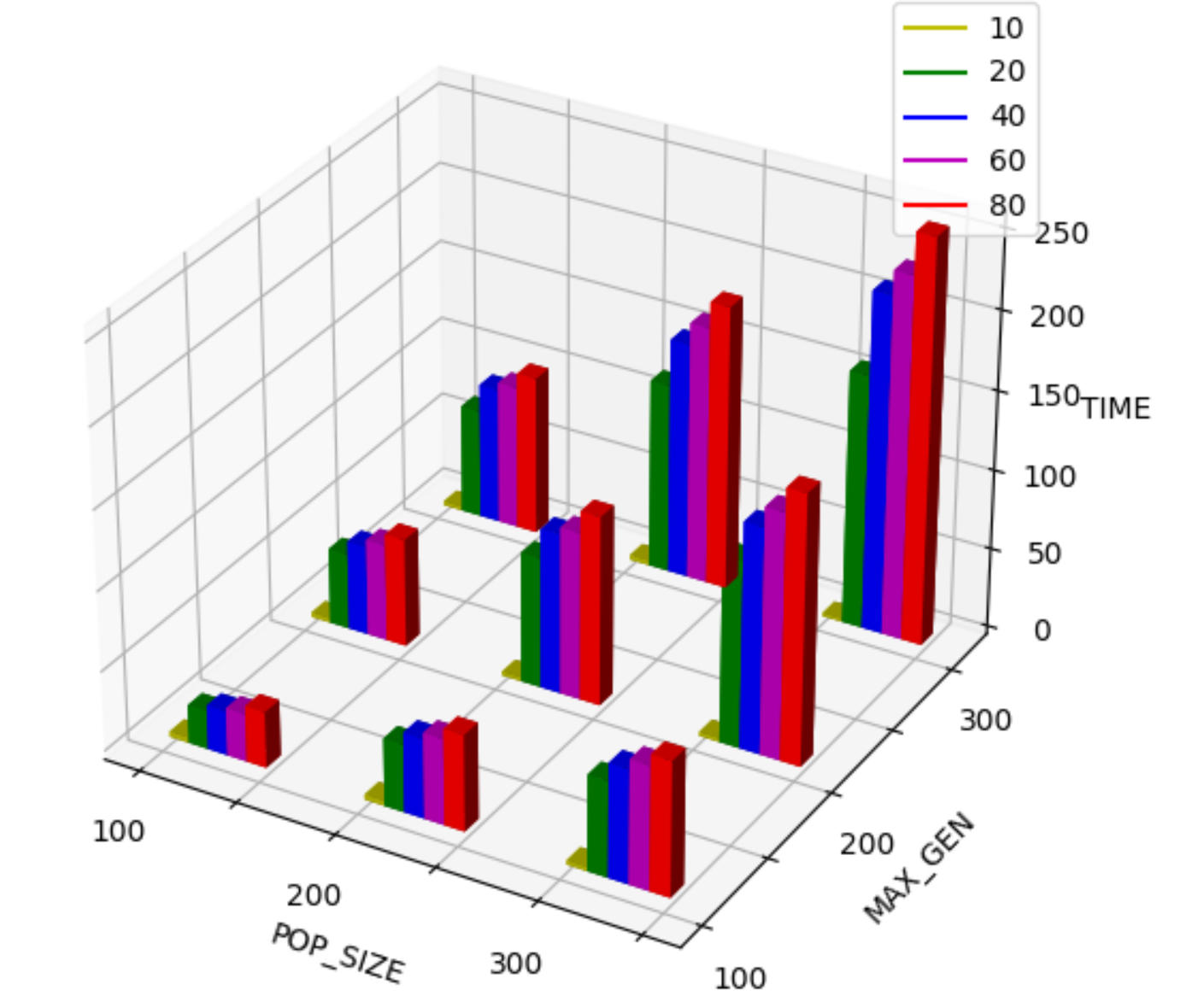
شکل5:نمودار زمان مصرفی بر حسب اندازه‌ی جمعیت و حداکثر تعداد نسل و اندازه‌ی مسئله (با رنگ جدا شده است) برای مسئله‌ی one\_max

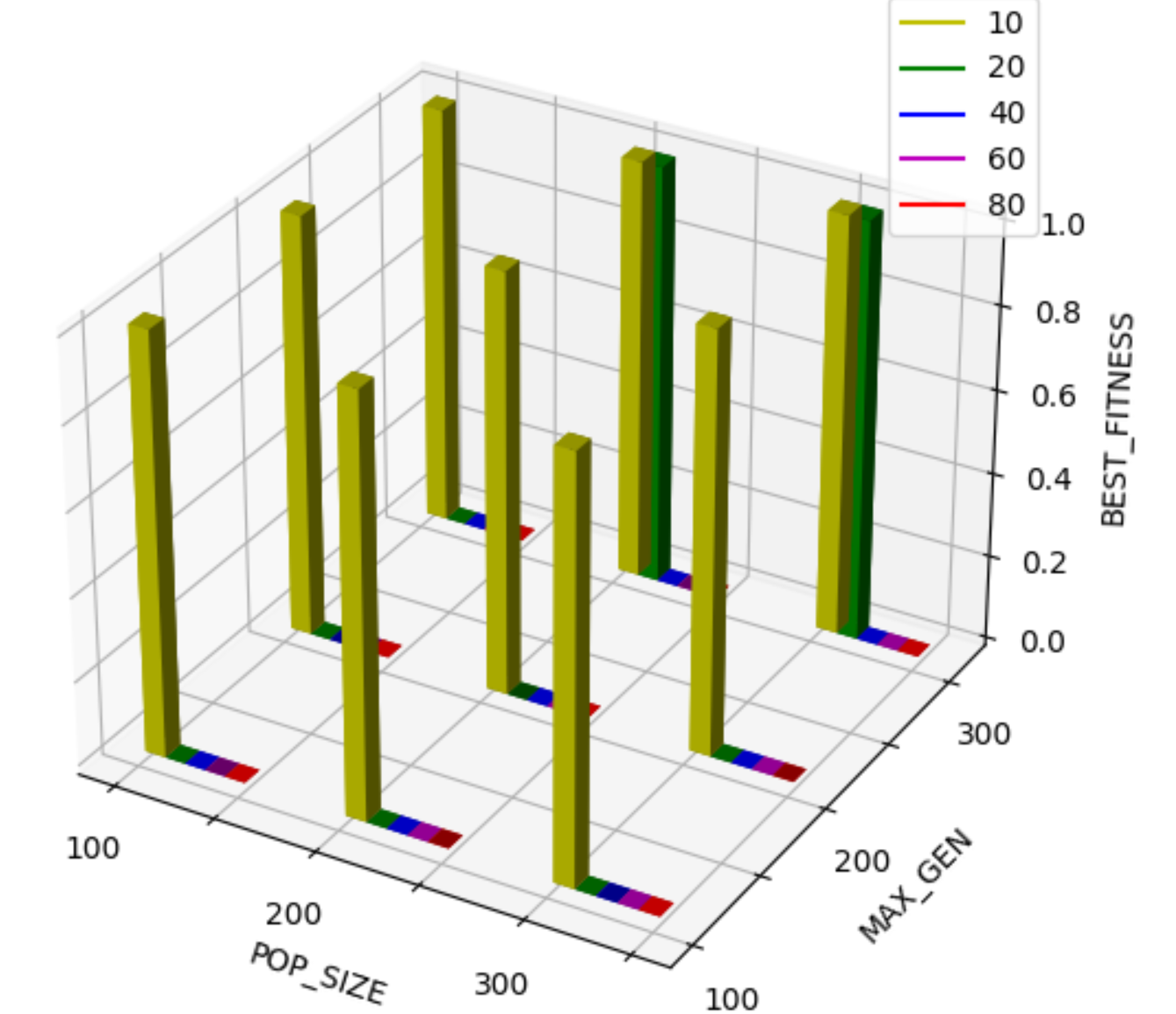
شکل3:نمودار میانگین برازندگی بر حسب اندازه‌ی جمعیت و حداکثر تعداد نسل و اندازه‌ی مسئله (با رنگ جدا شده است) برای مسئله‌ی one\_max

شکل4:نمودار بهترین برازندگی بر حسب اندازه‌ی جمعیت و حداکثر تعداد نسل و اندازه‌ی مسئله (با رنگ جدا شده است) برای مسئله‌ی one\_max

این نمودار‌ها برای مسئله‌ی Peakبه صورت شکل‌های 6، 7، 8 نشان داده شده است.





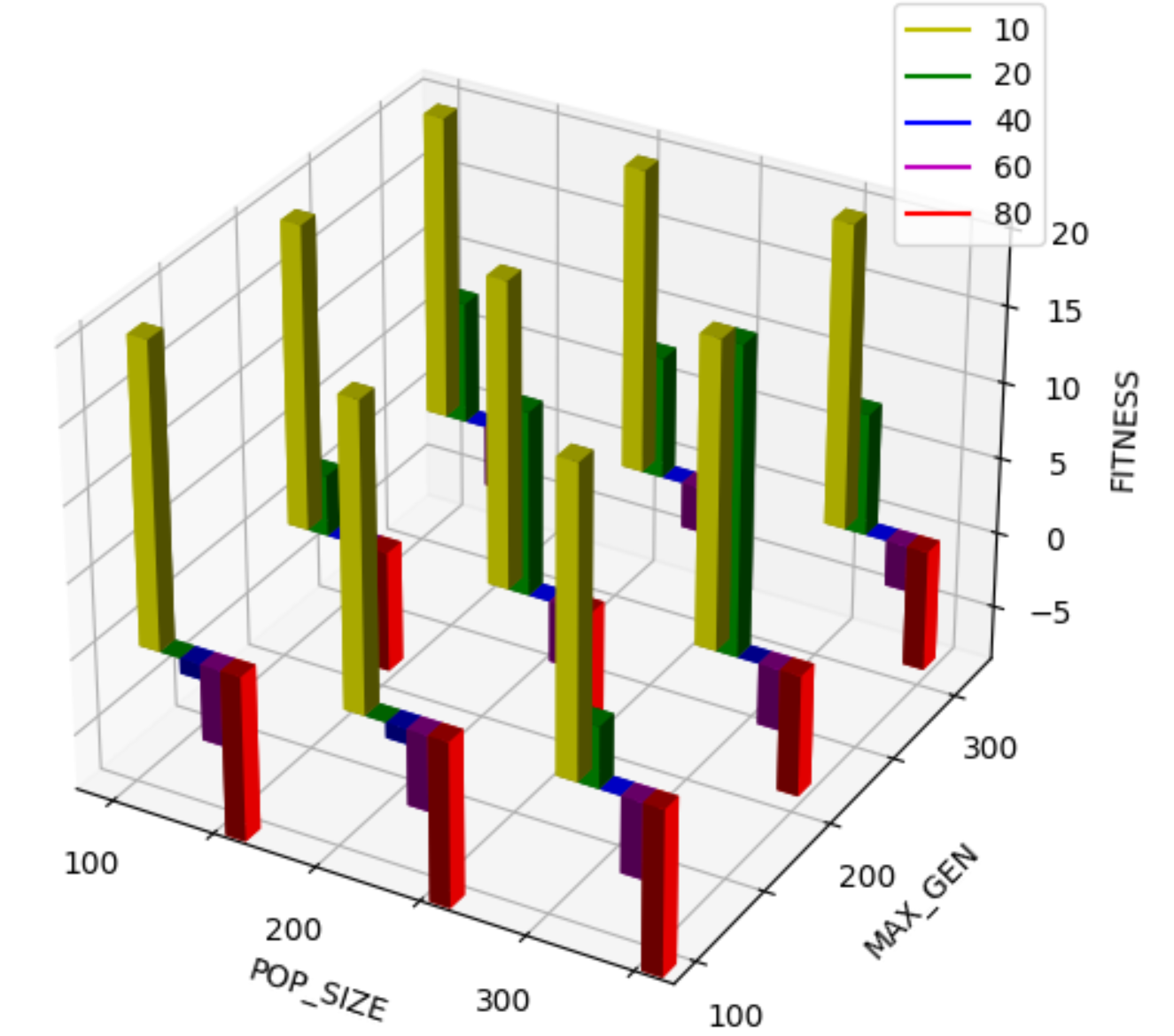


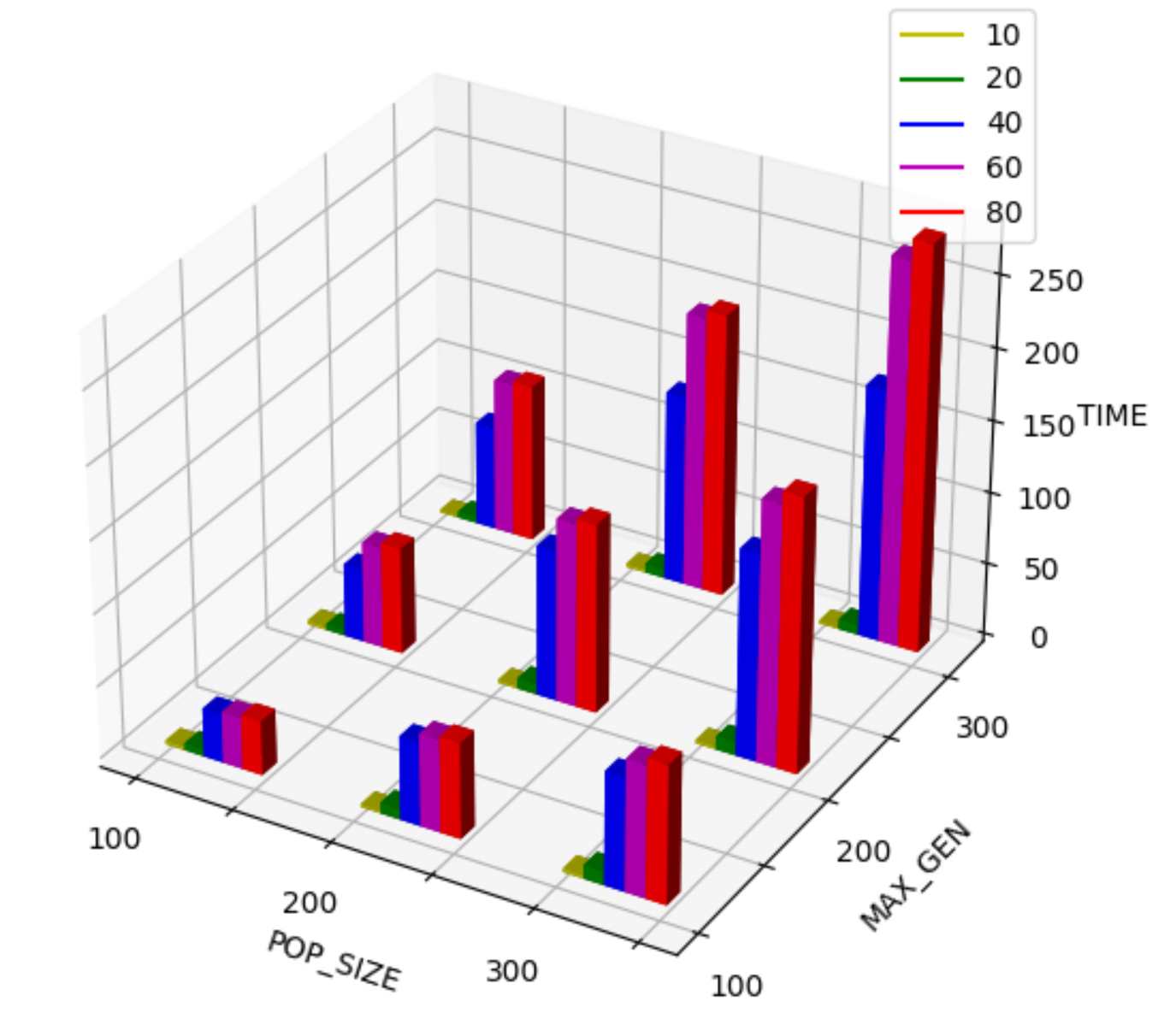
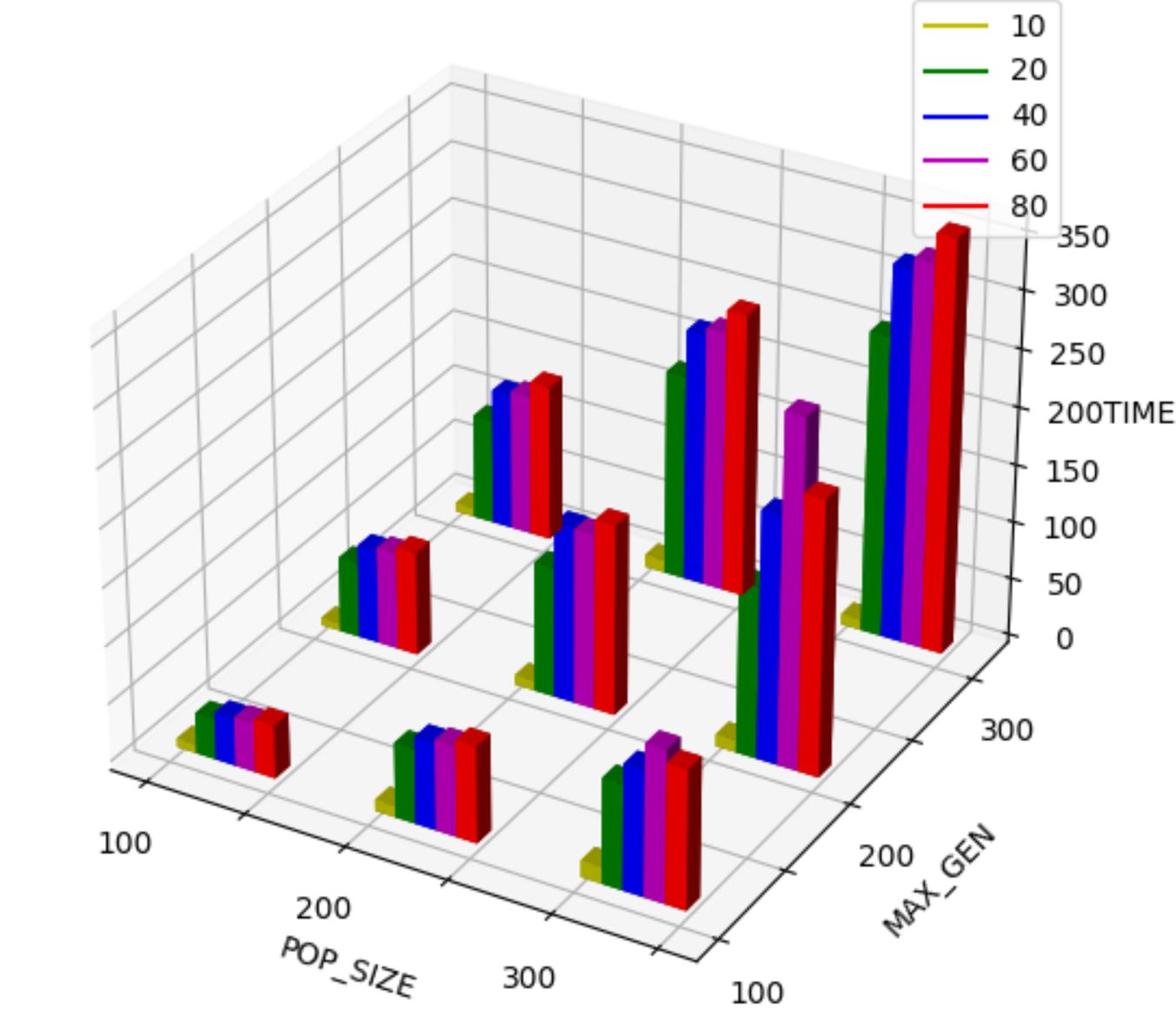
شکل6:نمودار میانگین برازندگی بر حسب اندازه‌ی جمعیت و حداکثر تعداد نسل و اندازه‌ی مسئله (با رنگ جدا شده است) برای مسئله‌ی peak

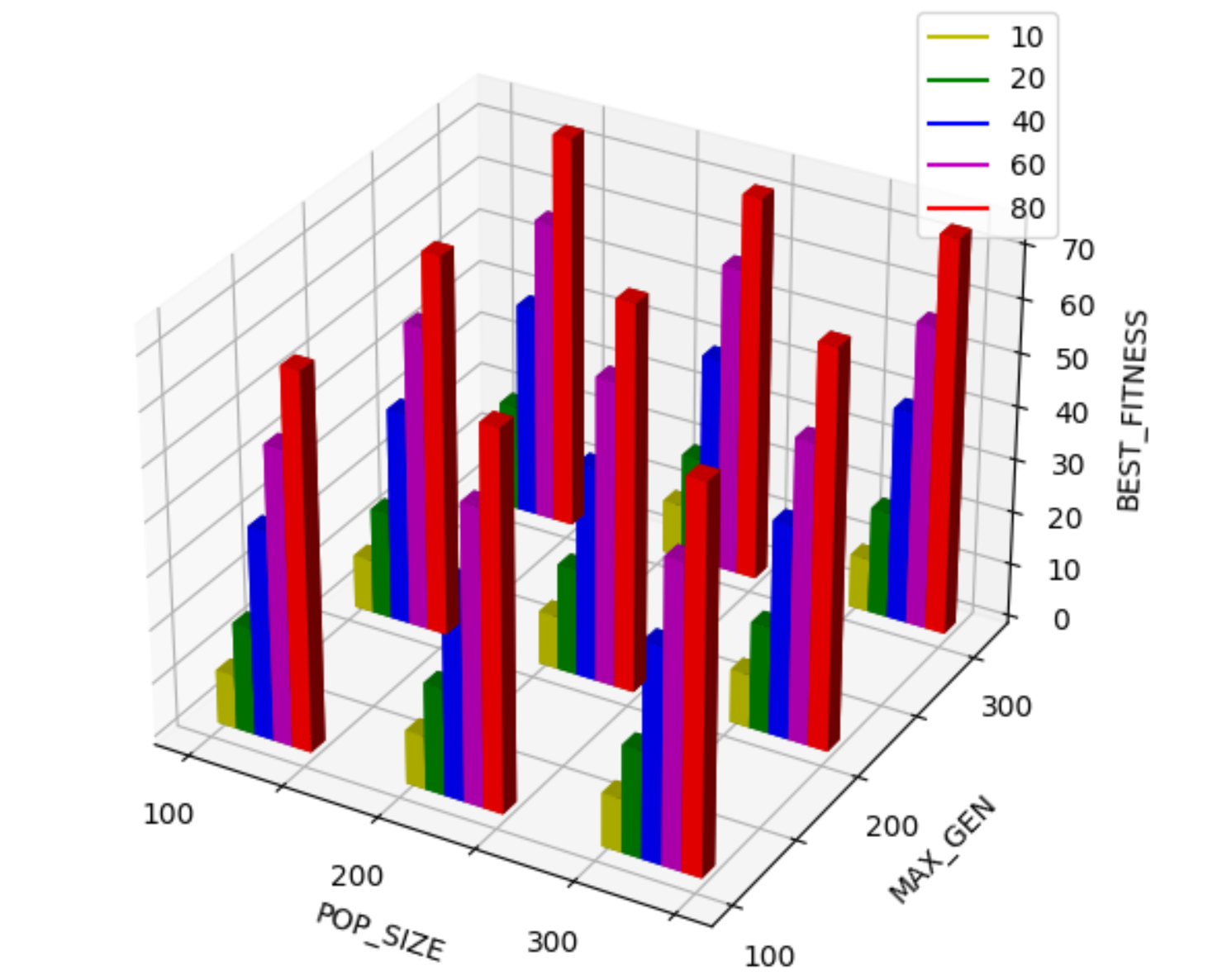
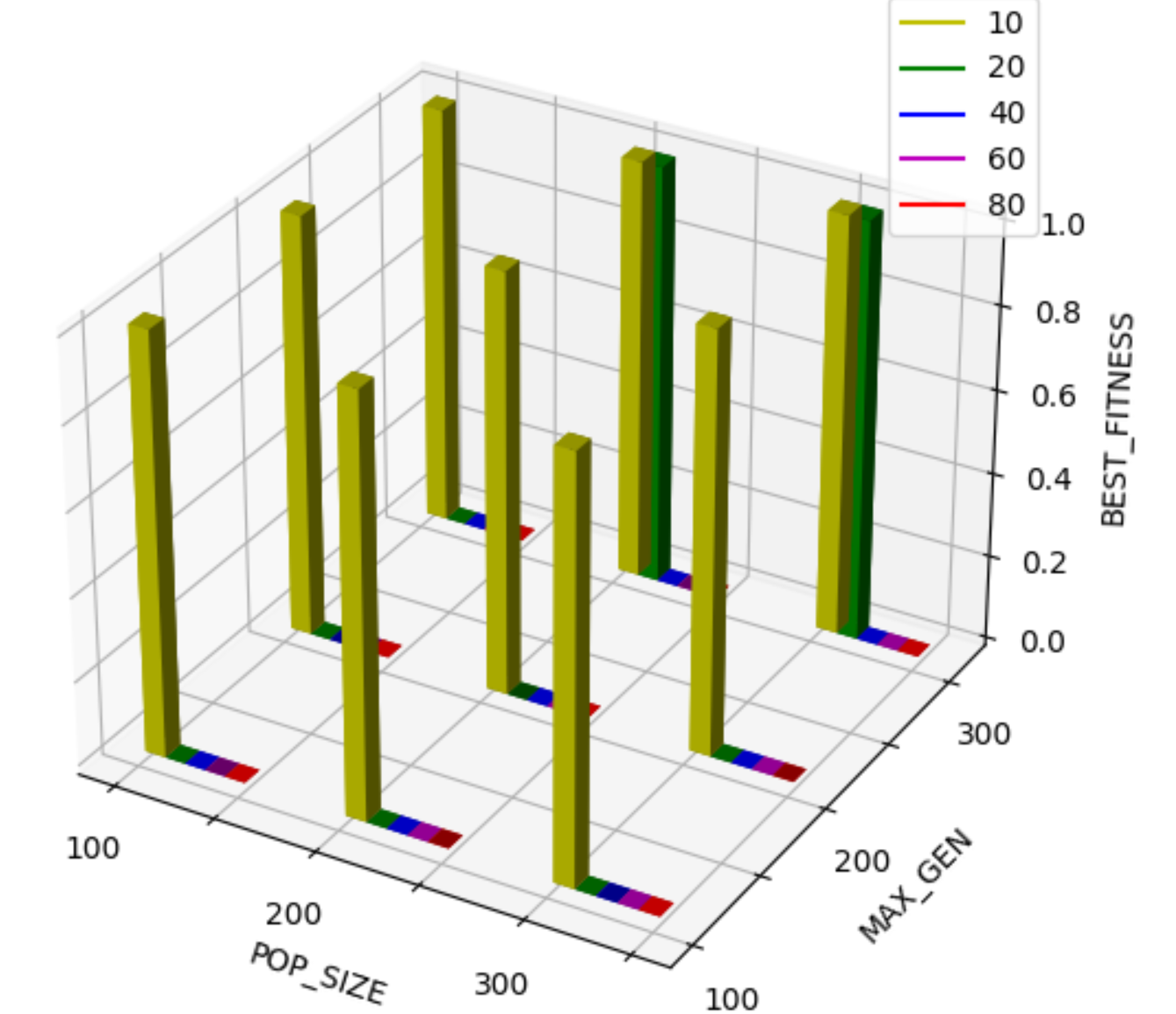
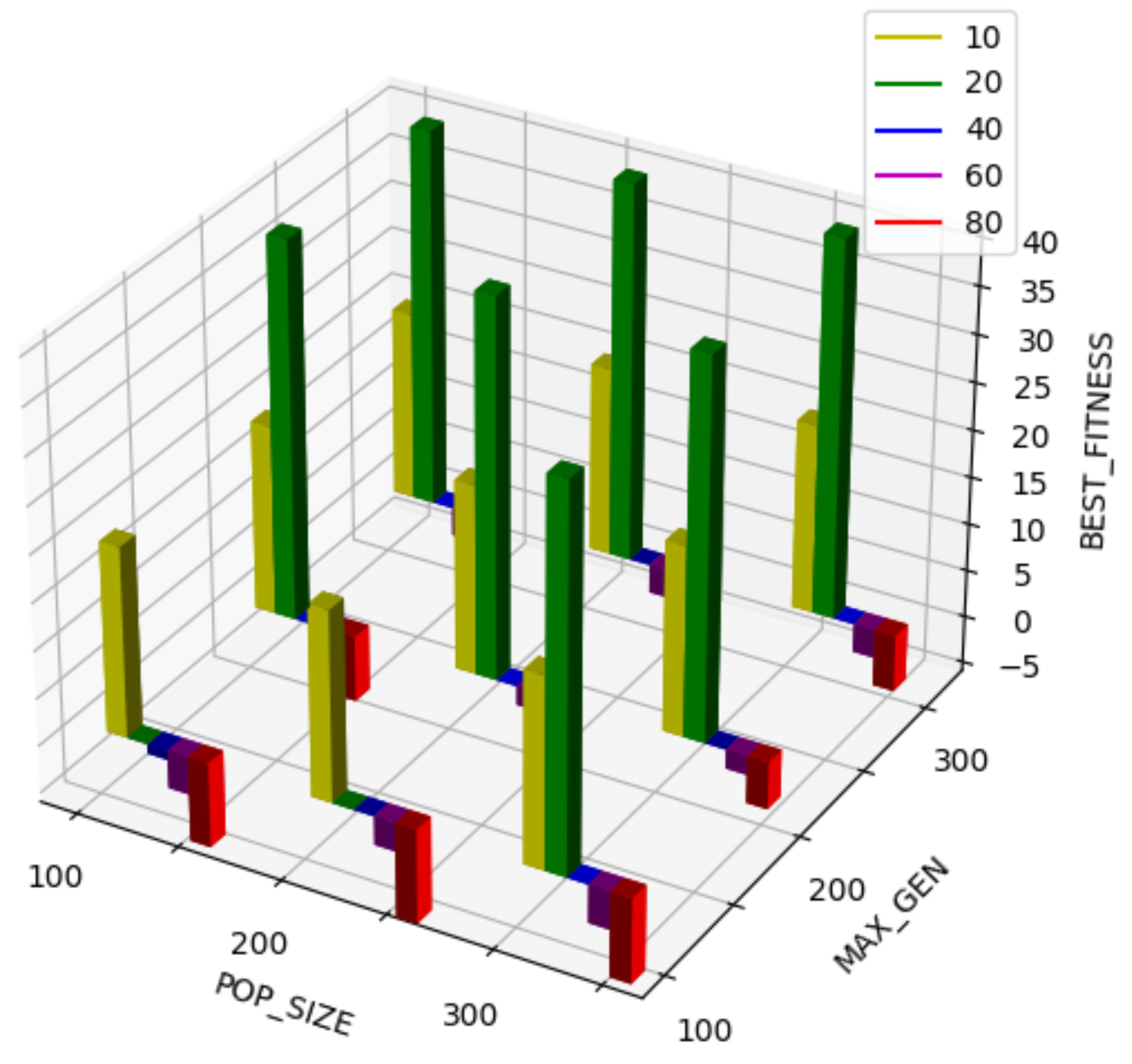
شکل7:نمودار بهترین برازندگی بر حسب اندازه‌ی جمعیت و حداکثر تعداد نسل و اندازه‌ی مسئله (با رنگ جدا شده است) برای مسئله‌ی peak

شکل8:نمودار زمان مصرفی بر حسب اندازه‌ی جمعیت و حداکثر تعداد نسل و اندازه‌ی مسئله (با رنگ جدا شده است) برای مسئله‌ی peak

این نمودار‌ها نیز برای مسئله‌ی Trapبه صورت شکل‌های 9، 10، 11 نشان داده شده است.





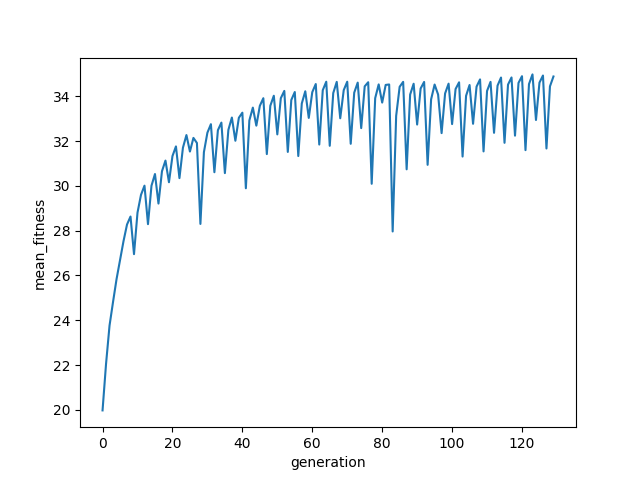


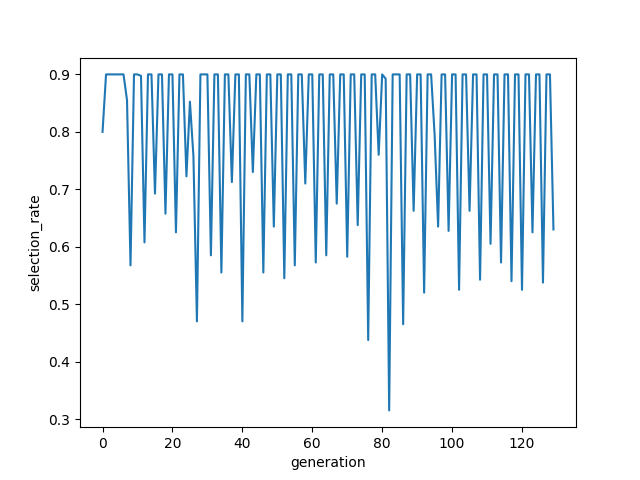
شکل11:نمودار زمان مصرفی بر حسب اندازه‌ی جمعیت و حداکثر تعداد نسل و اندازه‌ی مسئله (با رنگ جدا شده است) برای مسئله‌ی trap

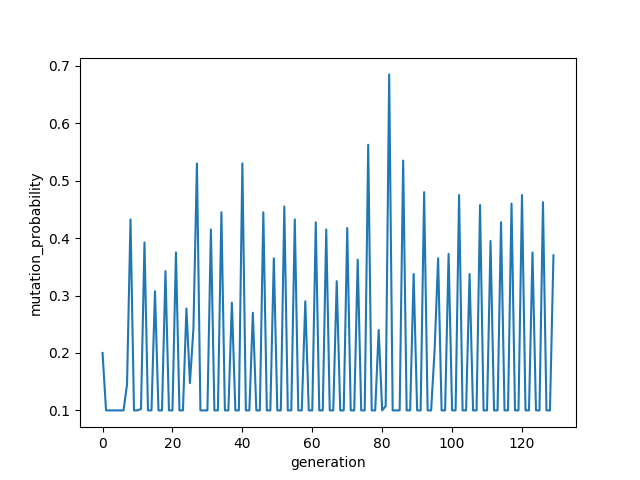
شکل9:نمودار میانگین برازندگی بر حسب اندازه‌ی جمعیت و حداکثر تعداد نسل و اندازه‌ی مسئله (با رنگ جدا شده است) برای مسئله‌ی trap

شکل10:نمودار بهترین برازندگی بر حسب اندازه‌ی جمعیت و حداکثر تعداد نسل و اندازه‌ی مسئله (با رنگ جدا شده است) برای مسئله‌ی trap

همچنین نمودار‌های میانگین برازندگی، نرخ فشار انتخاب و احتمال جهش در هر نسل نیز به دست آمده است که برای مثال برای اندازه‌ی مسئله‌ی 40 و حداکثر تعداد نسل 300 و اندازه‌ی جمعیت 300 برای سه مسئله به صورت شکل‌های 12 تا 20 آمده است.



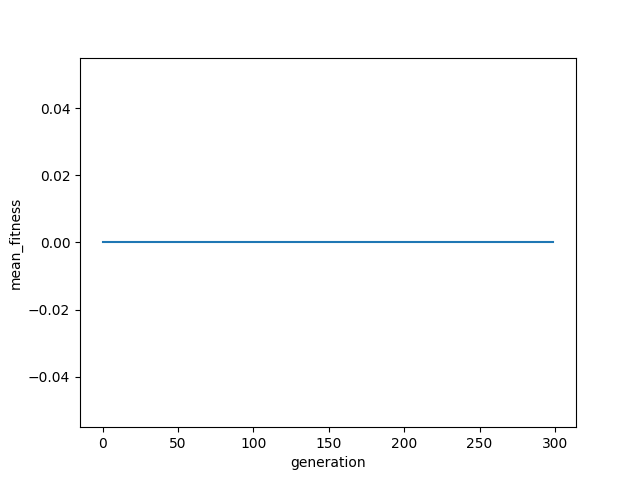


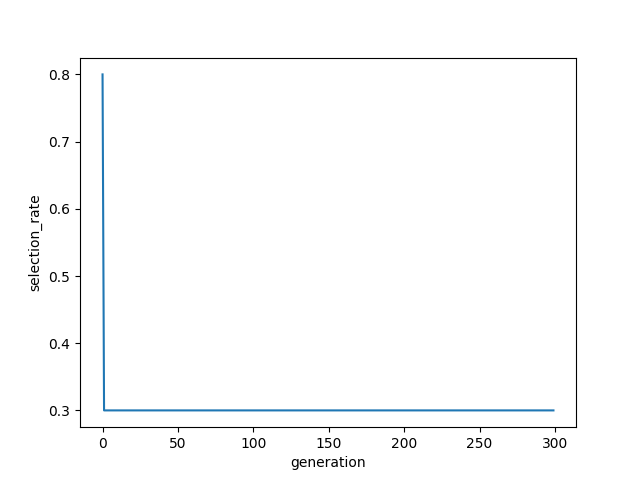


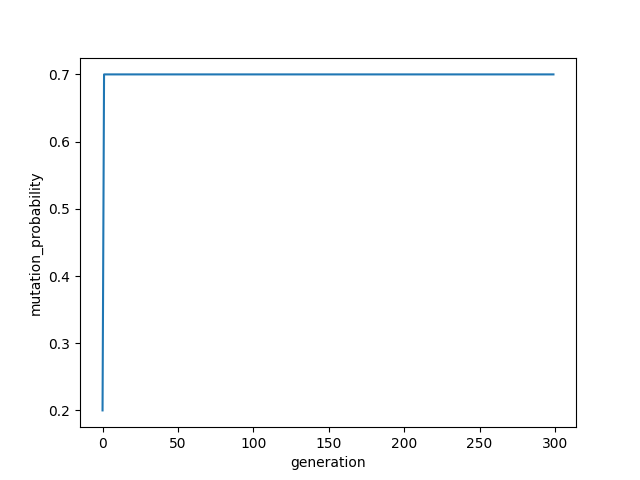
شکل14:نمودار تغییر نرخ فشار انتخاب در طی نسل ها برای اندازه‌ی مسئله‌ی 40 و حداکثر تعداد نسل 300 و اندازه‌ی جمعیت 300 برای تابع برازندگی one\_max

شکل12:نمودار میانگین برازندگی در طی نسل ها برای اندازه‌ی مسئله‌ی 40 و حداکثر تعداد نسل 300 و اندازه‌ی جمعیت 300 برای تابع برازندگی one\_max

شکل13:نمودار تغییر احتمال جهش در طی نسل ها برای اندازه‌ی مسئله‌ی 40 و حداکثر تعداد نسل 300 و اندازه‌ی جمعیت 300 برای تابع برازندگی one\_max



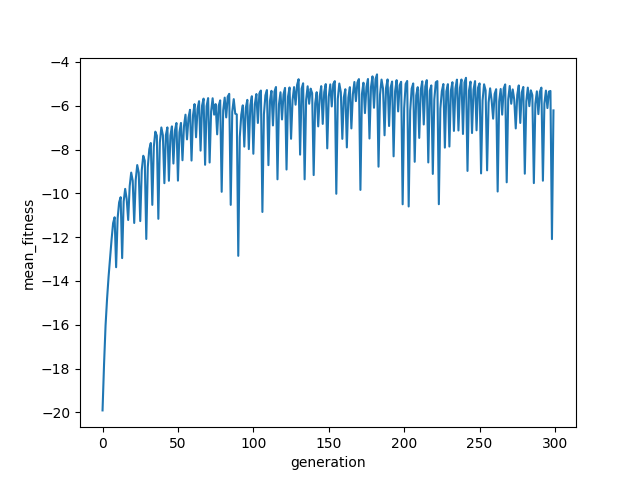


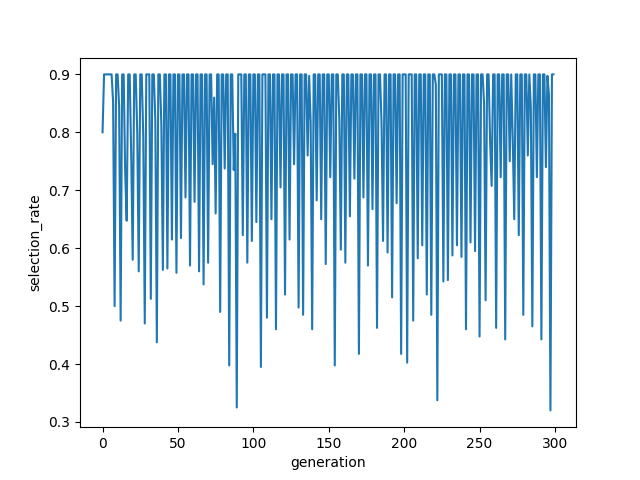


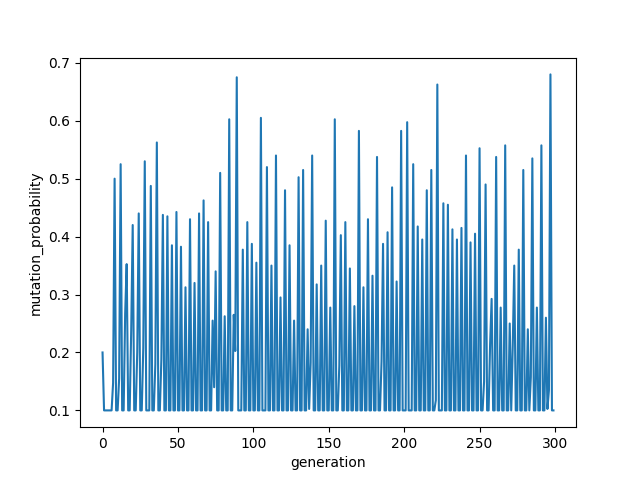
شکل17:نمودار تغییر نرخ فشار انتخاب در طی نسل ها برای اندازه‌ی مسئله‌ی 40 و حداکثر تعداد نسل 300 و اندازه‌ی جمعیت 300 برای تابع برازندگی peak

شکل 15:نمودار میانگین برازندگی در طی نسل ها برای اندازه‌ی مسئله‌ی 40 و حداکثر تعداد نسل 300 و اندازه‌ی جمعیت 300 برای تابع برازندگی peak

شکل 16: نمودار تغییر احتمال جهش در طی نسل ها برای اندازه‌ی مسئله‌ی 40 و حداکثر تعداد نسل 300 و اندازه‌ی جمعیت 300 برای تابع برازندگی peak







شکل18:نمودار میانگین برازندگی در طی نسل ها برای اندازه‌ی مسئله‌ی 40 و حداکثر تعداد نسل 300 و اندازه‌ی جمعیت 300 برای تابع برازندگی trap

شکل19:نمودار تغییر احتمال جهش در طی نسل ها برای اندازه‌ی مسئله‌ی 40 و حداکثر تعداد نسل 300 و اندازه‌ی جمعیت 300 برای تابع برازندگی trap

شکل20:نمودار تغییر نرخ فشار انتخاب در طی نسل ها برای اندازه‌ی مسئله‌ی 40 و حداکثر تعداد نسل 300 و اندازه‌ی جمعیت 300 برای تابع برازندگی trap

# 4-تحلیل الگوریتم

از نتایج به دست آمده از آزمایش، می‌توان فهمید که این الگوریتم نیز تا حد زیادی به اندازه‌ی جمعیت وابسته است و باید متناسب با اندازه‌ی مسئله این امر انتخاب شود و هر چه اندازه‌ی مسئله بزرگتر شد باید اندازه‌ی جمعیت نیز بزرگ‌تر شود. همچنین برای اندازه‌های مسئله‌ی بزرگ نیازمند ادامه پیدا کردن روند الگوریتم در نسل‌های زیاد، هستیم تا به جواب بهتر برسیم. از نتایج به دست آمده برداشت می‌شود که میزان زمان مصرفی به شدت وابسته به اندازه مسئله و اندازه‌ی جمعیت است و با زیاد شدن این دو مقدار، زمان مصرفی نیز به شدت زیاد می‌شود. بدیهی است که میزان زمان مصرفی نیز وابسته به حداکثر تعداد نسل نیز است. در مقایسه این الگوریتم با الگوریتم ژنتیک استاندارد که در گزارش قبل ذکر شده بود، به این نتیجه میرسیم که این الگوریتم کارایی بیشتری دارد و از لحاظ رسیدن به نقطه‌ی بهینه موفقیت بیشتری داشته است. همچنین برای مسائل peak به دلیل مساوی بودن برازندگی همه‌ی رشته‌ها به غیر از نقطه‌ی بهینه همواره این الگوریتم رویکرد کاوشی دارد و به سمت بهره‌وری نمی‌رود. همین امر در مسئله‌ی peak نیز مشاهده می‌شود با این تفاوت که به نقطه‌ی خوبی همگرا می‌شود و پس از آن به سمت اکتشاف به امید پیدا کردن نقطه‌ی بهینه حرکت می‌کند. از لحاظ زمان مصرفی در تعداد ثابت نسل، الگوریتم ژنتیک استاندارد، زمان کمتری مصرف می‌کند نسبت به این روش ولی مقدار آن تفاوت زیادی نداد و با توجه به نتایج به دست آمده میزان بهره‌وری این الگوریتم جدید به نسبت زمان مصرفی بهتر از الگوریتم ژنتیک است . همچنین میانگین برازندگی در کل جمعیت در هر نسل متفاوت است و ممکن است کم یا زیاد شود ولی در حالت کلی روند صعودی داشته است.

این الگوریتم حتی توانسته است برای مسائل trap و peak نیز در مسائل با اندازه‌ی 20 و بالاتر به جواب بهینه برسد و مسئله‌ی one\_max تا اندازه‌ی مسئله‌ی 40 نیز به نقطه‌ی بهینه رسیده است و در اندازه‌های بالاتر نیز به میزان برازندگی بسیار بالایی دست یافته است. الگوریتم trap نیز در بیشتر مواقعی که به نقطه‌ی بهینه (رشته‌ی تمام 1) نرسیده است ولی به نقطه‌ی برتر دوم یعنی رشته‌ی تمام صفر و در بقیه‌ی موارد به نقاط با صفر زیاد رسیده است.

تحلیل‌های به‌دست آمده از الگوریتم ژنتیک استاندارد در گزارش اول، در این الگوریتم جدید نیز صادق است.

# 5-کد مربوطه و خروجی‌های دیگر

روند الگوریتم که قبلا ذکر شده به طور کامل به صورت کد در قالب فایل ECA.py آمده و اسم توابع متناسب با عملکرد آن ذکر شده است. همچنین با استفاده از کدهای داخل فایل plot.py نمودار‌ها ترسیم شده است. همچنین نتایج دقیق به دست آمده در فایل‌های PDF به نام هریک از توابع در قالب جداول نمایش داده شده و برای هر مسئله با مؤلفه‌های متفاوت نمودارهای بدست آمده در بخش قبلی در نتایج بدست آمده قرار داده شده است.

# 6\_نتیجه گیری

از آزمایش‌های انجام شده، نتایج خوبی به دست آمده است که این به آن دلیل است که این الگوریتم، هم به جنبه‌ی اکتشاف و هم جنبه‌ی بهره‌وری توجه ویژ‌ه ای کرده است.

همچنان این الگوریتم جای پیشرفت دارد و می‌توان به جای در نظر گرفتن دو فرزند از دو والد تعداد بیشتر فرزند تولید کرد و احتمال موفقیت الگوریتم را بالا برد. همچنین می‌توان به به هر فرد متناسب با برازندگی یک مقدار طول عمر تخصیص داد تا هر فرد پس از تمام شدن طول عمرش از بین برود و در این حین اقدام به بازتولید می‌کرد و پیش‌بینی می‌شود که نتایج بهتری دهد. راهکار‌های بسیار زیاد دیگر نیز برای بهبود این الگوریتم وجود دارد.

این الگوریتم بسیار وابسته به طول جمعیت و حداکثر تعداد نسل است و اگر دو مقدار زیاد برای این دو در نظر گرفته شود، راه‌حل‌های بسیار خوبی به دست می‌آورد.