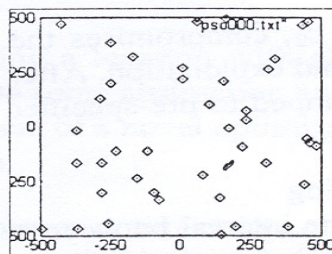


باسمه تعالی

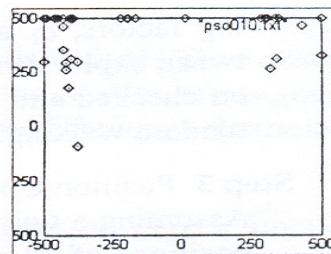
پروژه ۱: پرواز دسته‌جمعی پرندگان (Particle Swarm) ۴ نمره‌ای

انجام این پروژه به هیچ‌گونه دانش ریاضی یا درسی غیر از برنامه‌نویسی نیازی ندارد و با مطالب مطرح شده در کلاس قابل انجام است. توضیحات بیان شده فقط به منظور افزایش معلومات عمومی شماست.

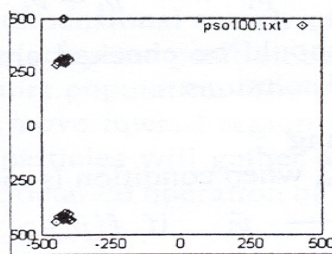
صورت مسأله: برنامه‌ای بنویسید که ابتدا ۲۰ نقطه را به طور تصادفی در صفحه نشان دهد. در هر مرحله، هر نقطه بر اساس فرمولی که در ادامه می‌آید، مقداری جابجا می‌شود. کاربر باید بتواند با فشردن دکمه سمت راست، مرحله بعد را ببیند (بر اساس فرمول جابجایی که گفته می‌شود نقاط را جابجا کنید). در هر مرحله فقط نقاط را نشان دهید.



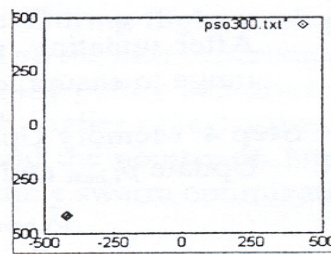
(a) 0-th iteration.



(b) 10-th iteration.



(c) 100-th iteration.



(d) 300-th iteration.

همچنین برای سادگی، نیازی نیست که بیش از ۵۰۰ مرحله وجود داشته باشد (۵۰۰ مرحله کافی است).

برای سادگی بیشتر، محدوده متغیرها، تابع و نیز بردار جابجایی مشخص و ثابت در نظر می‌گیریم:

فرض کنید مقدار x و y هر نقطه، عددی صحیح بین ۰ تا ۳۰۰ است.

تابع $z=f(x,y)$ را به صورت زیر در نظر بگیرید:

$$z = x \cdot \sin(\sqrt{|2x|}) + y \cdot \sin(\sqrt{|2y|})$$

بردار جابجایی:

در مرحله اول، هر کدام از نقاط به تصادف جهتی را انتخاب می‌کنند و در آن جهت حرکت می‌کنند. اندازه بردار حرکت نباید زیاد باشد (مثلا حداکثر ۱۰ واحد به سمت چپ یا راست و حداکثر ۱۰ واحد به سمت بالا یا پایین). در مراحل بعد، بردار جابجایی برای هر نقطه از قانون زیر بدست می‌آید:

$$0.5d + 0.3R_1(q - p) + 0.2R_2(g - p)$$

که در آن R_1 و R_2 دو عدد تصادفی از صفر تا یک است. d بردار جابجایی قبلی است. p مکان فعلی نقطه است. q مکانی است که این نقطه در مراحل قبل، بیشترین مقدار تابع را در آنجا داشته است. g مکانی است که بیشترین مقدار تابع توسط مجموعه همه نقاط، در مراحل قبل، در آنجا پیدا شده است.

امکانات اجباری (برای گروههای ۱ نفره و ۲ نفره):

- کاربر باید بتواند با هر بار فشردن دکمه سمت چپ، یک مرحله به عقب برگردد.
- کاربر باید بتواند با فشردن دکمه s در هر مرحله‌ای که خواست، اطلاعات آن مرحله را در فایلی ذخیره کند. بدین منظور، یک فایل متنی ایجاد کنید و در هر سطر از فایل، به ترتیب مقدار x ، y و z یک نقطه را بنویسید.
- برای آنکه تجمع نقاط معلوم باشد، یک نقطه تکی را با یک pixel مثلا سفید نشان دهید. اگر چند نقطه روی هم افتادند رنگ آن pixel را زرد کنید و اگر تعداد نقاطی که روی هم افتادند خیلی زیاد شد (بیش از 1/4 کل نقاط) از رنگ قرمز استفاده کنید.

امکانات اجباری فقط برای گروههای دونفره:

- در ابتدای برنامه، محدوده مجاز از کاربر پرسیده شود (x حداقل و حداکثر و y حداقل و حداکثر) و نقاط اولیه در آن محدوده انتخاب شوند و نمایش داده شوند. برای سادگی فرض کنید طول و عرض محدوده مجاز بیشتر از همان ۳۰۰ نخواهد بود ولی ممکن است به جای 0 تا 300، مثلا در محدوده 200- تا 100 یا مثلا 400 تا 550 باشد.

- چنانچه طول یا عرض محدوده نمایش کمتر از ۳۰ واحد بود، دقت را ۱۰ برابر کنید. مثلا اگر محدوده مجاز x از 0 تا 25 وارد شد به جای اعداد صحیح برای x ، اعداد 0, 0.1, 0.2, ..., 24.9, 25 (هر 0.1 یک pixel)
- کاربر بتواند انتخاب کند که نقاط اولیه تصادفی باشند یا از یک فایل خوانده شوند. فایل از نوع رکورد زیر است و آدرس فایل توسط کاربر داده می‌شود:

Type Trec = Record

x , y : Real;

End;

امکانات اختیاری:

- تعداد نقاط از کاربر پرسیده شود (حداکثر ۵۰ نقطه).
- کاربر باید بتواند با هر بار فشردن دکمه‌های بالا و پایین، ۱۰ مرحله به جلو یا عقب برود (حرکت سریع).
- شماره مرحله و مقدار ماکزیمم در آن مرحله (ماکزیمم نقاط آن مرحله)، همواره نمایش داده شود.

توضیحات اضافی: برخی از حیوانات رفتارهای اجتماعی جالبی دارند. این رفتارها به آنها امکان می‌دهد توانایی خود را افزایش داده، در یک حرکت گروهی، کارهای بزرگی انجام دهند. به عنوان مثال به حرکت دسته‌جمعی ماهی‌ها یا پرواز دسته‌جمعی پرندگان وقتی به دنبال غذا می‌گردند دقت کنید. ایده تقریبی از این قرار است که در ابتدا هر کدام به طور جداگانه در منطقه‌ای به دنبال غذا می‌گردند. وقتی که مقداری غذا پیدا شد، اطرافیان هم با خبر می‌شوند و برخی از آنها به سمت غذا می‌روند. وقتی که غذای زیادی پیدا شد همه به دور آن جمع می‌شوند. در واقع اگر تعداد زیادی ماهی یا پرنده یکجا جمع شدند معلوم می‌شود آنجا غذای زیادی وجود دارد.

از ایده فوق برای پیدا کردن ماکزیمم یک تابع استفاده می‌شود. فرض کنید یک تابع دو متغیره $z=f(x,y)$ داریم که می‌خواهیم حداکثر آن را بیابیم. در واقع این تابع، مقدار غذا را در هر نقطه نشان می‌دهد (اگر قرار بود این تابع رسم شود یک شکل سه‌بعدی بدست می‌آمد؛ به ازای هر (x,y) در صفحه، ارتفاع شکل برابر با z است؛ هرچا شکل بلندتر بود ماکزیمم است؛ با این حال نیازی به رسم آن نیست و ما با یک شکل دوبعدی سر و کار داریم

نه سه بعدی). چند نقطه به تصادف در صفحه انتخاب می‌کنیم (مانند پرندگان که در ابتدا هرکدام به یک گوشه می‌روند). مقدار تابع را در هر نقطه پیدا می‌کنیم (فرض کنید یک function در برنامه‌تان نوشته شده که با گرفتن دو ورودی x و y ، خروجی z را پیدا می‌کند). این مقدار، نشانگر مقدار غذایی است که یک پرنده در آن نقطه پیدا می‌کند. در مرحله بعد، بر اساس فرمولی که گفته می‌شود، هر پرنده مقداری جابجا می‌شود تا تعدادی نقطه جدید بدست آید و دوباره همان مراحل تکرار می‌شود (محاسبه مقدار تابع، جابجایی، ...). پس از چند مرحله، همه نقاط یکسان خواهند شد (همه پرندگان یکجا جمع می‌شوند؛ در واقع این نقطه ماکزیمم است).