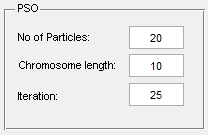
Blunder detection via Particle Swarm Optimization

در این روش پس از دریافت پارامتر های مورد نظر از کاربر (مطابق شکل ذیل) برداری به نام Particle\_Length که طولی به اندازه تعداد Particle دارد و هر ژن آن حاوی یک عدد در بازه 1 تا طول کروموزوم است تشکیل می شود. این بردار تعداد نقطه ای که در هر Solution جای می گیرد را نشان می دهد.



سپس جمعیت اولیه بر اساس این بردار تشکیل می شود.

تابع برازش در این روش بدین صورت است که سطحی Bilinear به همه نقاط یک Patch به غیر از نقاط داخل کروموزوم برازش داده می شود و بردار V (باقیمانده) برای نقاط شرکت کننده در تقریب کمترین مربعات به دست می آید. مجموع توان دوم همه المان های بردار V،Cost هر solution را ارائه می کند. پس از مرتب کردن Solution ها براساس cost آن ها به صورت نزولی، بهترین Solution به عنوان Gbest ذخیره می شود و طول این Solution در متغیر Best\_Particle\_Length ذخیره می گردد. (تابع Cost\_Fnc مقدار هزینه را برمیگرداند.)

شروع حلقه تکرار در تابع اصلی:

برنامه وارد حلقه تکرار می شود. در این حلقه کلیه Solution ها به همراه بردار Particle\_Length وارد تابع NeigborPoint\_Fnc می شوند.

ورود به تابع NeigborPoint\_Fnc:

در این تابع یک بردار Lbest که متناظر با طول مشخصی از کروموزوم هاست ساخته می شود. مثلا Lbest مربوط به طول 2 بهترین Solution ای که دارای طول 2 است را در خود ذخیره می کند.

در این حلقه هر Solution ورودی بررسی می شود یعنی ابتدا بردار V که در بالا توضیح داده شد محاسبه می شود و بر حسب این که طول آن Solution چند است به همان تعداد از نقاطی که بیشترین باقیمانده را در بردار V دارند در کروموزوم جدید جای می گیرند. سپس Cost این کروموزوم محاسبه شده اگر از المان متناظر خودش در Lbest (هم طول با خودش) بهتر بود جایگزین می شود و در غیر این صورت رها می شود و Solution بعدی بررسی می گردد. این کار تا پایان همه Solution های ورودی ادامه می یابد. در پایان جمعیت جدید از بهترین های ذخیره شده در Lbest بر حسب بردار Particle\_Length به وجود می آید.

بازگشت به تابع اصلی:

پس از بازگشت به تابع اصلی، Cost همه Solution ها محاسبه می شود و بردار جمعیت و به تبع آن بردار Particle\_Length مرتب می گردند. حال مقایسه Gbest با بهترین Solution این تکرار صورت می گیرد. اگرSolution بهتری یافت شده بود جایگزین می شود و طول آن نیز در متغیر Best\_Particle\_Length ذخیره می شود.

در ادامه نوبت به بروز سازی بردار Particle\_Length است. در یک حلقه تک تک طول ها با Best\_Particle\_Length مقایسه می شود. اگر کوچکتر از Best\_Particle\_Length بود به مقدار Particle\_Length یکی اضافه می شود و اگر بزرگتر بود یکی کم می شود. در این جا یک دور حلقه تکرار پایان می یابد. این کار تا پایان تعداد تکرار مشخص شده توسط کاربر ادامه می یابد و نهایتا Gbest بهترین جواب را ارائه می کند.