فصل یکم

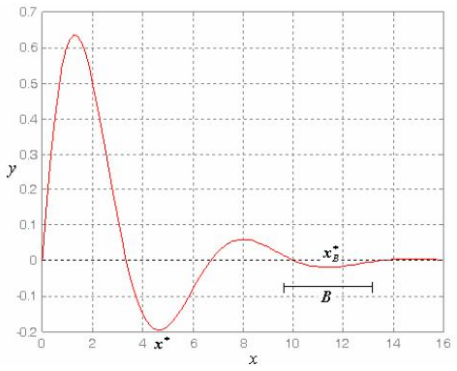
بهینه‌سازی دستهٔ ذرات

۱-۱- تعریف بهینه سازی

بهینه‌سازی بدین مفهوم است که در بین پارامترهای یک تابع به دنبال مقادیری باشیم که تابع را کمینه یا بیشینه می‌نمایند. کلیه مقادیر مناسب جهت این امر را، راه حلهای ممکن[1] و بهترین مقدار از این مقادیر را راه حل بهینه[2] می‌نامند. الگوریتم‌های بهینه‌سازی هر دو نوع مسائل بیشینه‌سازی و کمینه‌سازی را پوشش می‌دهند. به این علت که هر مسأله بیشینه‌سازی قادر به تبدیل به مسائل کمینه‌سازی می‌باشد. یک مسأله کمینه‌سازی به صورت زیر تعریف می‌گردد :

$f: S \rightarrow R$ که $S \subseteq RNd$ و Nd ابعاد فضای جستجوی S می‌باشد. x^* کمینه کننده عمومی نامیده می‌شود و$f(x)$ کمینه عمومی[3] نامیده می‌شود. x_B^* کمینه کننده محلی نامیده میشود کمینه عمومی و محلی در شکل ۰-۱ نمایش داده شده است.

$$f(x^*) \le f(x), \forall x \in S, x^* \in S$$
$$f(x_B^*) \le f(x), \forall x \in B \quad B \subset S \qquad (۱)$$



شکل ۰-۱) : مثالی از کمینه عمومی x^* و کمینه محلی x_B^*

بهینه سازی کاربردهای زیادی دارد از جمله در تخصیص منابع، زمان بندی‌ها، تصمیم گیری‌ها و ..... روشهای مختلفی برای بهینه سازی وجود دارد. البته در دنیای طبیعی مسائلی وجود دارند که چند تابع را در عین واحد بهینه می‌نمایند. این مسائل multiobjective نامیده می‌شوند.

یافتن راه حل برای مسائلی که از نوع NP می‌باشند بسیار مشکل است. الگوریتم‌هایی مانند جستجوی اتفاقی تا حدی این مشکل را حل نموده اند. توسط این نوع جستجوها راه حل‌هایی پیدا می‌شوند که تقریباً به جواب

[1] Feasible solution
[2] Optimization solution
[3] Global minima

نزدیکند. از جمله این الگوریتم ها می‌توان به تپه‌نوردی[۱] ، شبیه‌سازی تابکاری‌فلزات[1] [۲] و جستجویTabu [۳] و الگوریتم‌های تکاملی و بهینه سازی گروه ذرات اشاره نمود. [۴]

۱-۲- بهینه سازی گروه ذرات[2]

۱-۲-۱- تاریخچه بهینه سازی گروه ذرات

در بیشتر گونه‌های جانوران رفتارهای گروهی دیده شده است. چه بسا که بعضی از این گونه‌ها نیز توسط یک عضو برتر گروه راهنمایی می شوند. به عنوان مثال در شیرها، میمونها و گوزنها این امر کاملا مشاهده می‌شود. در اوایل سال ۱۹۰۰ با تحقیقاتی که بر روی رفتار اجتماعی میمونها صورت گرفت مشخص شد که در این گونه از میمونها عملکرد و رفتار هر عضو از گروه به صورت سلسله مراتبی از طرف جامعه بالاتر، فرمان داده می‌شود. مطلب جالب تری که وجود دارد این است که گونه هایی از جانوران وجود دارند که به صورت گروهی زندگی می‌نمایند اما راهنمایی ندارند. هر عضو یک رفتار خود سازمانده۳ دارد که بدون استفاده از یک راهنما می‌تواند در محیط حرکت نموده و نیازهای طبیعی خود را بر طرف نماید. مانند گروه پرندگان، ماهی‌ها و گله گوسفندان. این گونه از جانوران هیچ دانشی نسبت به رفتار عمومی کل گروه ندارند و یا حتی هیچ دانشی نسبت به محیطی که در آن قرار دارند، ندارند. در عوض قادرند با رد و بدل نمودن اطلاعات با اعضای همجوار خود در محیط حرکت نمایند. این تعامل ساده بین ذرات باعث ایجاد رفتار پیچیده تر گروه می‌شود. مانند جستجوی یک محیط توسط ذرات. تحقیقات زیادی بر روی رفتارهای اجتماعی ذرات انجام شده است که در ادامه به چند نمونه از آنها می‌پردازیم:

رفتار پرندگان

رفتار گروه ماهی‌ها

رفتار شکار کردن وال‌های گوژپشت

رفتار جستجوی غذا در میمونهای وحشی

رفتار اظهار عشق و جستجوی غذا در کوسه ها

جهت فهم بهتر رفتار پویای گروه ذرات، در[۵]بعضی از رفتارها شبیه سازی شدند و رفتار اعضای گروه به دسته های زیر تقسیم شدند:

---

[1] Simulated annealing

[2] Particle Swarm Optimization

[33] Self -Organizing

اجتناب از برخورد[1]: اعضای یک گروه با یکدیگر برخوردی ندارند.

تنظیم سرعت[2]: هر عضو سرعت خود را متناسب با اعضای همسایه خود تنظیم می‌نمایند.

جمع شدن مرکزی[3]: هر عضو تلاش می‌کند که در کنار همسایگان خود حرکت نماید.

در بسیاری از کاربردهای بهینه‌سازی از این شبیه سازی‌ها الهام گرفته شده است. در ادامه به مدل خاصی از بهینه سازی ها که از زندگی جانوران انبوه الهام گرفته است پرداخته میشود که بهینه سازی گروه ذرات[2] یا PSO نام دارند.

اولین بار کندی و ابرهارت پس از شبیه سازی رفتار اجتماعی پرندگان روش بهینه سازی گروه ذرات را ارائه دادند.[۶] اجزای یک گروه از یک رفتار ساده تبعیت می‌نمایند. بدین نحو که هر عضو از گروه از موفقیت سایر همسایگانشان تقلید می نماید. هدف از این الگوریتم ها این است که اعضای گروه در فضای جستجو حرکت نموده و در یک نقطه بهینه (مانند منبع غذا) جمع شوند. این رفتار شبیه فرضیه ای است که در [۷]ارائه شده است. روش PSO ریشه در کارهای Reynolds دارد که یک شبیه سازی ابتدایی از رفتار اجتماعی پرندگان است. در این مدل رفتارهای ساده پیدا کردن نزدیکترین همسایه‌ها (تنظیم سرعت ها) پیاده شده است. این مدل پرندگان به صورت تصادفی در یک فضای جستجوی جدول پیکسلی قرار داده می‌شوند و در هر تکرار نزدیکترین همسایه ذره انتخاب شده و سرعت ذره با سرعت نزدیکترین همسایه‌اش جایگزین می‌شود. این عمل باعث می‌گردد که گروه خیلی سریع به یک جهت حرکت نامعین و بدون تغییر همگرا شوند. جهت رفع این مشکل یک مولفه دیوانگی به صورت تغییر تصادفی در گروه ها استفاده شده است.

به منظور توسعه بیشتر این مدل مفهوم سردسته پرندگان[4] به مدل اضافه گردید که به شکل یک حافظه از بهترین موقعیتهای هر عضو و همسایگان آن عضو بود. بهترین موقعیت قبلی هر عضو بهترین موقعیتی است که آن عضو از ابتدای حیات خود تا بحال کسب نموده است. بهترین موقعیت همسایگی بهترین موقعیتی است که توسط همسایگان یک عضو ملاقات شده است. این دو بهترین موقعیت به عنوان نقاط جذب عمل می نمایند. با استفاده از یک مجموعه قوانین ساده می توان موقعیت‌های اعضای گروه را به روز نمود. بدین صورت که عضو به یک نسبت به سمت دو موقعیت بهتر حرکت می نماید. به مرور زمان با تکرار الگوریتم اعضا حول یک هدف جمع می شوند. این رفتار حتی بدون هماهنگی سرعتها و فاکتور دیوانگی نتیجه بخش بود. مدل نهایی **بهینه‌ساری گروه ذرات** نامیده شد. به جای کلمه اعضا از کلمه **ذره**[5] استفاده شد گر چه ذرات بدون وزن و حجم هستند یک سرعت و شتاب به

---

[1] Collision avoidasnce
[2] Velocity matching
[3] Flock Centering
[4] rooster
[5] particle

هر کدام از آنها نسبت داده می شود. کلمه **گروه**[1] اساسا بدلیل حذف هماهنگی سرعتها (که یکی از اجزای رفتار گروهی است) استفاده شده است و با تعاریف Milonas نیز هم خوانی دارد. اصول پایه ای هوش گروهی در این مرجع به صورت زیر تعریف شده است.

اصل هم جواری[2]: گروه عناصر باید قادر باشند که محاسبات را در فضا و زمان اندکی انجام دهند

اصل کیفیت[3] : گروه ذرات باید قادر باشند که به فاکتورهای کیفی محیط پاسخ دهند.

اصل تنوع پاسخ[4]: ذرات نباید فعالیتش را بر روی کانالهای خیلی باریک محدود نماید.

اصل پایداری[5]: گروه ذرات نباید حالت رفتار خود را هر بار با تغییرات محیط تغییر دهد.

اصل تطابق[6]: گروه ذرات باید قادر باشد.`رفتار خودر ار به نفع کاهش هزینه مجاسباتی تغییر دهد.

بهینه سازی گروه ذرات به صورت زیر این موضوعات را بر آورده می نماید که محاسبات فضای چند بعدی به صورت یک سری از گامهای زمانی انجام می شوند. که این اصل اول را پوشش می‌دهد. گروه ذرات به فاکتورهای کیفی به صورت بهترین موقعیتهای فردی و همسایگی جواب می‌دهد. تخصیص پاسخها بین بهترین موقعیت ملاقات شده ذره و بهترین موقعیت ملاقات شده گروه تنوع پاسخها را تضمین می‌نماید. گروه حالت خود را فقط هنگامیکه بهترین موقعیت ملاقات شده توسط ذره و بهترین موقعیت ملاقات شده توسط گروه تغییر می‌کنند، تغییر می‌دهد. که اصل پایداری را پوشش می‌دهد. در نهایت گروه رفتار تطبیقی از خود نشان می‌دهد بدین صورت که حالت خود را هنگامیکه بهترین موقعیت ملاقات شده توسط ذره و بهترین موقعیت ملاقات شده توسط گروه تغییر می کنند، تغییر می‌دهد.

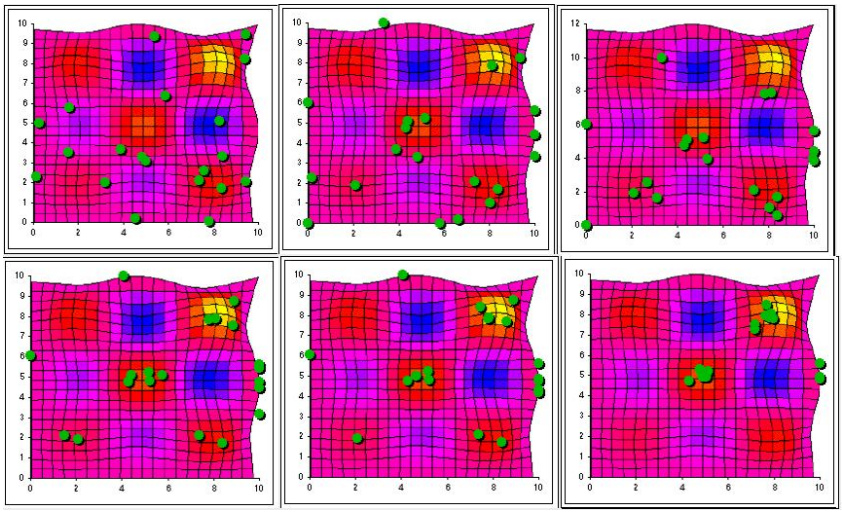
۱-۲-۲- الگوریتم بهینه سازی گروه ذرات

روش PSO یک روش سراسری بهینه‌سازی است که با استفاده از آن می‌توان با مسائلی که جواب آنها یک نقطه یا سطح در فضای n بعدی می‌باشد، برخورد نمود. در اینچنین فضایی، فرضیاتی مطرح می‌شود و یکسرعت ابتدایی به ذرات اختصاص داده می‌شود، همچنین کانال‌های ارتباطی بین ذرات درنظر گرفته می‌شود. سپس این ذرات در فضای پاسخ حرکت می‌کنند، و نتایج حاصله بر مبنای یک «ملاک شایستگی» پس از هر بازهٔ زمانی محاسبه می‌شود. با گذشت زمان، ذرات به سمت ذراتی که دارای ملاک شایستگی بالاتری هستند و در گروه

---

[1] swarm
[2] Proximity Principle
[3] Quality Principle
[4] Principle of Diverse response
[5] Principle of Stability
[6] Principle of Adaptebility

ارتباطی یکسانی قرار دارند، شتاب می‌گیرند. مزیت اصلی این روش بر استراتژی‌های بهینه سازی دیگر این است که، تعداد فراوان ذرات ازدحام کننده، باعث انعطاف روش در برابر مشکل پاسخ بهینهٔ محلی می‌گردد. در شکل ۲-۰) : نمونه هایی از روند حرکت ذرات در فضای جستجو نمایش داده شده است. عکس موجود در گوشه بالا و سمت چپ تصویر موقعیت اولیه ذرات را نشان می هد که در فضای جستجوی دو بعدی قرار دارندو با تکرارهای الگوریتم در نهایت ذرات به صورت عکس موجود در گوشه پایین سمت راست تصویر همگرا می شوند.



شکل ۲-۰) : روند حرکت ذرات در یک گروه.

هر ذره دارای یک موقعیت است که مشخص می‌نماید مختصات ذره در فضای جستجوی چند بعدی چه می باشد باحرکت ذره در طول زمان موقعیت ذره تغییر می‌نماید. $x_i(t)$ موقعیت ذره iام در زمان tام را مشخص می‌نماید. همچنین هر ذره برای حرکت نمودن در فضا نیاز به یک سرعت دارد $v_i(t)$ سرعت ذره iم در زمان tام را مشخص می نماید. با افزودن سرعت به موقعیت هر ذره، می توان موقعیت جدیدی برای ذره در نظر گرفت. معادله به روز نمودن موقعیت ذره در(۲)آورده شده است.

$$x_i(t+1) = x_i(t) + v_i(t+1)$$
$$x_i(t) \sim U(x_{\min}, x_{\max}) \qquad (۲)$$

اینکه موقعیت یک ذره در فضای جستجو موقعیت مناسبی است یا خیر توسط یک تابع شایستگی ارزیابی می‌گردد. ذرات توانایی این را دارند که بهترین موقعیتی را که در طول حیات خود در آن قرار داشته‌اند به خاطر بسپارند. به بهترین تجربه فردی یک ذره یا بهترین موقعیت ملاقات شده توسط ذره y_i گفته می‌شود (در بعضی ار الگوریتم ها y_i به عنوان pbest نیز نام گذاری شده است) و ذرات می‌توانند از بهترین موقعیت ملاقات شده توسط کل گروه نیز آگاهی داشته باشند. که این موقعیت $\hat{y}_i$ نامیده می‌شود. (در بعضی ار الگوریتم ها $\hat{y}_i$ به عنوان gbest نیز نام گذاری شده است) بردار سرعت ذره در فرایند بهینه سازی منعکس کننده دانش تجربی ذره و اطلاعات جامعه ذرات است. هر ذره برای حرکت در فضای جستجو دو مولفه را مد نظر دارد:

مؤلفه شناختی[1]: $y_i(t)-x_i(t)$ بهترین راه حلی است که یک ذره به تنهایی بدست می‌آورد.

مؤلفه اجتماعی[2]: $\hat{y}_i(t)-x_i(t)$ بهترین راه حلی است که توسط کل گروه تشخیص داده می‌شود.

دو مدل اصلی برای الگوریتم PSO استاندارد وجود دارد که محاسبه بردار سرعتشان بر اساس دو مولفه شناختی و اجتماعی می‌باشد. این دو مدل به نامهای lbest PSO وgbest PSO می‌باشند که تفاوت آنها در سایز همسایگی است که برای هر ذره در نظر گرفته می‌شود. که در ادامه به آنها خواهیم پرداخت.

۱-۲-۲-۱- مدل Global Best PSO

در این الگوریتم همسایه یک ذره کل ذرات گروه می‌باشد. توپولوژیی که برای اتصال ذرات به یکدیگر نیاز است شبیه توپولوژی ستاره است $\hat{y}$ در این الگوریتم بهترین موقعیت ملاقات شده توسط کل گروه می‌باشد. موقعیت هر ذره توسط فرمول(۳) ارزیابی می‌گردد. (f تابع ملاک شایستگی مربوط به ذرات می باشد.)

$$+1)=\begin{cases} y_i(t)\ \ if\ \ f(x_i(t+1))\geq f(y_i(t)) \\ x_i(t+1)\ if\ \ f(x_i(t+1))\leq f(y_i(t)) \end{cases} \qquad (۳)$$

بردار $\hat{y}$ به صورت فرمول(۴)محاسبه می‌گردد.

$$\in\{y_0,y_1,...,y_s\}=\min\{f(y_0(t)),f(y_1(t)),....,f(y_s(t))\} \qquad (۴)$$

S سایز گروه (تعداد ذرات) را مشخص می‌نماید. ذرات در ابتدا در فضای جستجو پراکنده‌اند و سپس هر ذره با سرعت خاصی به سمت یک نقطه همگرا می‌شود. سرعت یک بردار N_d بعدی است که V_{ij} سرعت j امین عنصر از بردار سرعت ذره i را نمایش می‌دهد. بنابراین سرعت ذره i بر اساس فرمول(۵)بهنگام می‌گردد.

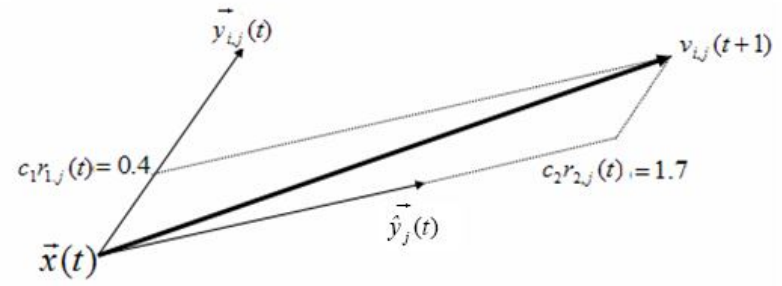
$$+1)=wv_{i,j}(t)+c_1r_{1,j}(t)(y_{i,j}(t)-x_{i,j}(t))+c_2r_{2,j}(t)(\hat{y}_j(t)-x_{i,j}(t))$$
$$,....,N_d \qquad (۵)$$

w وزن اینرسی[3] را مشخص می‌نمایند و c_1 c_2 ثابتها می باشند.$r_{1j}(t)\ ,\ r_{2j}(t)\ \approx U(0,1)$

عبارت وزن اینرسی برای اولین بار توسط شای و ابرهارت در سال ۱۹۹۸ معرفی گردید [۸] این وزن در واقع درصدی از سرعت قبلی ذره را در محاسبه سرعت جدید تاثیر می‌دهد. هر چه این مقدار بیشتر باشد جستجوی عمومی افزایش می‌یابد و هرچه این وزن کمتر باشد میزان جستجوی محلی افزایش می‌یابد. در شکل ۰-۳ نحوه به روز نمودن سرعت ذره نمایش داده شده است.

---

[1] Cognetive Component
[2] Social Component
[3] Inertia Weight



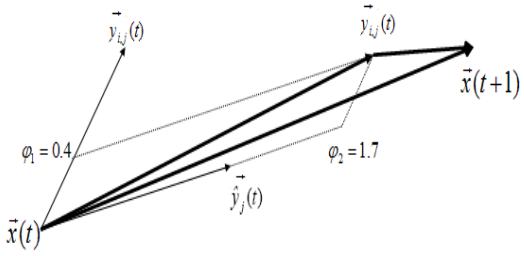
شکل ۰-۳) بهنگام سازی سرعت

باید یک ارتباطی بین مقادیر ثابت و وزن موجود باشد. تا رسیدن به جواب را تضمین نماید. چون در غیر این صورت باعث می‌شود که ذرات بر اثر یک سری از رفتارها واگرا شوند.

$$(c_1 + c_2)/2 - 1 \langle w_c \qquad (۶)$$

موقعیت ذره i توسط معادله(۷) بهنگام می‌گردد:

$$x_i(t+1) = x_i(t) + v_i(t+1) \qquad (۷)$$



شکل ۰-۴): بهنگام سازی موقعیت ذره

سرعت و موقعیت هر ذره توسط معادلات ذکر شده بهنگام می‌گردد و آنقدر این عمل تکرار می‌گردد تا یا به حداکثر تعداد تکرار برسد و یا سرعت بهنگام شده به صفر نزدیک شود و عملکرد هر ذره توسط ملاک شایستگی برآورد می‌شود.

۱-۲-۲-۲- مدل Local Best PSO:

در این مدل یک ذره جهت به روز نمودن سرعت خود فقط توانایی برقراری ارتباط با تعدادی ذره که در همسایگی آنها قرار دارند را دارد. Ni مجموعه ذرات همسایه یک ذره می باشند. توپولوژیی که برای ارتباط بین ذرات بکار برده می شود توپولوژی حلقه است. بهترین موقعیت ملاقات شده توسط همسایگان ذره $\hat{y}_i$ نامیده می شود. سایز همسایگان با l نمایش داده می‌شود.

$$N_i = \{y_{i-l}(t), y_{i-1+l}(t), ..., y_{i-1}(t), y_i(t), y_{i+1}(t), ..., y_{i+l-1}(t), y_{i+1}(t)\}$$
$$\hat{y}_i(t+1) \in \{N_i \mid f(\hat{y}_i(t+1)) = \min\{f(y_i(t))\}, \forall y_i \in N_i\} \quad (۸)$$
$$v_{i,j}(t+1) = wv_{i,j}(t) + c_1 r_{1,j}(t)(y_{i,j}(t) - x_{i,j}(t)) + c_2 r_{2,j}(t)(\hat{y}_j(t) - x_{i,j}(t))$$

معادله بهنگام سازی موقعیت ذره‌ها مانند روش قبل است و تغییری نمی‌نماید. همسایه‌های یک ذره عملاً مشخص کننده رفتارهای اجتماعی یک ذره هستند. لذا توپولوژی همسایه‌ها موضوع مهم و مفیدی می باشد. اگر l=s باشد lbest مانند gbest عمل می‌نماید. که در این صورت همسایه های یک ذره کل ذرات گروه می‌باشد. الگوریتم PSO استاندارد درشکل ۰-۵ نمایش داده شده است.

For each particle $i \in 1,...,s$ **do**

  Randomly initialize x_i

  Randomly initialize v_i (or just set v_i to zero)

  Set $y_i = x_i$

endfor

Repeat

  For each particle $i \in 1,...,s$ **do**

    Evaluate the fitness of particle i, $f(x_i)$

    Update y_i using $y_i(t+1) = \begin{cases} y_i(t) & \text{if } f(x_i(t+1)) \geq f(y_i(t)) \\ x_i(t+1) & \text{if } f(x_i(t+1)) < f(y_i(t)) \end{cases}$

    Update $\hat{y}$ using $\hat{y}(t) \in \{y_0, y_1, \ldots, y_s\} = \min\{f(y_0(t)), f(y_1(t)), \ldots, f(y_s(t))\}$

    For each dimension $j \in 1,...,N_d$ **do**

      Apply velocity update using

        $v_{i,j}(t+1) = wv_{i,j}(t) + c_1 r_{1,j}(t)(y_{i,j}(t) - x_{i,j}(t)) + c_2 r_{2,j}(t)(\hat{y}_j(t) - x_{i,j}(t))$

    endloop

    Apply position update using $x_i(t+1) = x_i(t) + v_i(t+1)$

  endloop

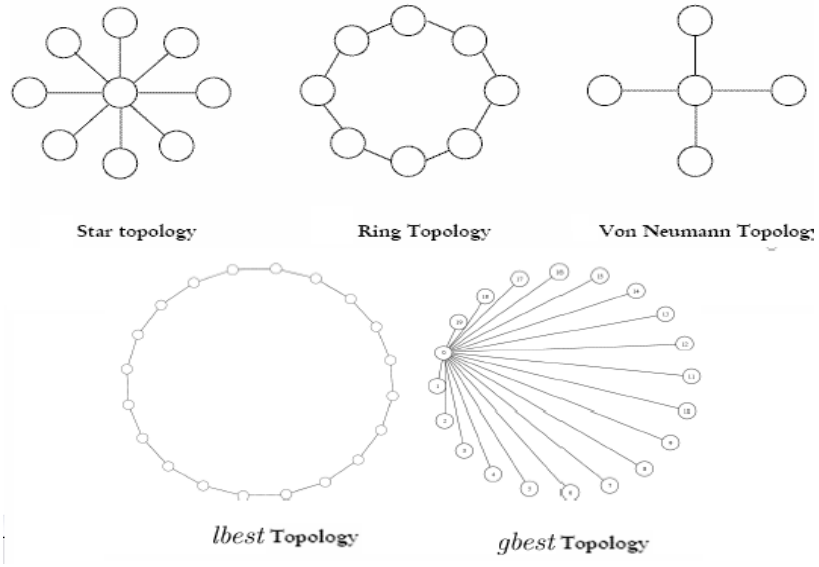
Until some convergence criteria is satisfied

شکل ۰-۵): الگوریتم عمومی PSO

۱-۲-۳- توپولوژی یا ساختار شبکه اجتماعی

برای همسایه‌های یک ذره می‌توان توپولوژی‌های متنوعی در نظر گرفت متداولترین توپولوژی‌هایی که استفاده می‌گردد، حلقه‌ای و ستاره‌ای می‌باشد. در توپولوژی ستاره یک ذره به عنوان hub یا مرکز در نظر گرفته می‌شود که کلیه ذرات گروه را باهم متصل می‌نماید. در واقع کلیه ذرات دیگر فقط به hub متصل هستند. در توپولوژی حلقه‌ای یا رینگ ذرات در یک حلقه قرار دارند و هر ذره دارای تعدادی همسایه در سمت چپ و تعدادی همسایه در سمت

راستش است. اخیراً مدل جدیدی از توپولوژی همسایه‌ها که بر اساس مدل وان نیومن است ارایه شده است. در این مدل ذرات با توجه به شبکه‌ای بهم متصلند.(شبکه های دو بعدی) هر ذره به ۴ تا از همسایگانش متصل است. در شکل ۰-۶ توپولوژی‌های مختلف نمایش داده شده اند. انتخاب توپولوژی تأثیر به سزایی بر یافتن بهترین راه حل گروه دارد. با استفاده از مدل gbest سرعت بسیار افزایش می یابد. (تمام ذره های گروه در تکرار t+1 ام از بهترین راه حلی که در تکرار t ام پیدا کرده اند تأثیر می گیرند). استفاده از توپولوژی وان نیومن و حلقه‌ای سرعت رسیدن به پاسخ را کاهش می دهند. به این دلیل که بهترین راه حل پیدا شده بر اساس تعداد زیادی از همسایگان حاصل می گردد که هنوز بر روی کل ذرات گروه تأثیر ندارند. این آهستگی رسیدن به جواب ذره ها را قادر می‌سازد تا نواحی بیشتری از فضا را مورد جستجو قرار دهند و از همگرایی زود رس جلوگیری می شود.



شکل ۰-۶) دیاگرام توپولوژی همسایه ها در PSO

۱-۲-۳-۱- بررسی مشکلات PSO

الگوریتم بهینه‌سازی گروه ذرات دارای چندین نقطه ضعف می‌باشد.در این الگوریتم احتمال قرار گرفتن ذرات در بهینه های محلی وجود دارد. هر چند که PSO نسبت به کلیه الگوریتم های تکاملی دارای سرعت بالاتری است اما معمولاً نمی‌تواند کیفیت رسیدن به راه حل را با افزایش تکرارها جبران کند. یکی از دلایل این است که در این الگوریتم ذرات به یک نقطه خاص که بین بهترین موقعیت عمومی و بهترین موقعیت شخصی قرار دارد همگرا می-شوند. به علت این نقطه ضعف تغییرات زیادی در PSO داده شده است. یکی از این تغییرات وزن اینرسی یا w می-باشد. نقطه ضعف دیگر وابستگی این روش به مسأله می‌باشد. این وابستگی معمولاً نتیجه تغییرات در تنظیم پارامترهای الگوریتم است. در کل نمی‌توان یک پارامتر را برای کلیه مسائل بکار برد. یکی از عیبهای عمده الگوریتم PSO استاندارد در زیر آورده شده است:

فرض شود که ذره i در گروه، دارای سرعت، موقعیت و بهترین موقعیت ملاقات شده باشد. هر ذره به تنهایی یک بردار n بُعدی را نمایش می‌دهد که معرف یک پاسخ یا راه حل برای مساله می‌باشد. گاهی امکان دارد که قسمت هایی از این بردار به پاسخ صحیح نزدیک شده باشند در حالیکه قسمتهای دیگر بردار از پاسخ صحیح دور باشند. بنابراین در کل این ذره مناسب به نظر نمی رسد و باید به موقعیت بهتری برود. امکان دارد که آن قسمتهایی از بردار ذره که به جواب نزدیک بوده‌اند در طی به روز نمودن موقعیت ذره جدید، از پاسخ جدید فاصله بگیرند بنابراین اطلاعات مفید ذره از بین می‌رود. برای روشن شدن مطلب به یک مثال زیر توجه نمایید. فرض شود که یک ذره با بردار ویژگی سه بُعدی وجود دارد که تابع ملاک شایستگی آن به صورت $F(x) = \|x-a\|^2$ می‌باشد.

که a=(20,20,20) است و *x مقدار کمینه عمومی[1] می باشد که برابر با a است. فرض شود که ذره با بردار x2 وجود دارد و $\hat{y}_i$ بهترین موقعیتی است که ذره تا بحال داشته است. اگر t مرحله جاری الگوریتم را نمایش دهد به احتمال زیاد داریم:

$$\| x_2(t+1) - \hat{y}(t+1) \| < \| x_2(t) - \hat{y}(t) \|$$

در صورتیکه فرض کنیم که $\hat{y}_i$ در طول این مرحله تغییر ننماید، در مرحله t+1 ذره x_2 به $\hat{y}_i$ نزدیکتر می شود. فرض شود که در مرحله t ذره x_2 و $\hat{y}_i$ دارای مقادیر $\hat{y}(t) = (17,2,17), x_2(t) = (5,20,5)$ باشند. تابع ملاک شایستگی f برای هر کدام از موارد بالا $f(\hat{y}(t)) = 342 \ \ and \ \ f(x_2(t)) = 450$ می باشد. فرض شود که در مرحله t+1 ذره x_2 و $\hat{y}_i$ دارای مقادیر $\hat{y}(t+1) = (17,2,17), x_2(t+1) = (15,5,15)$ باشند.

مقدار $x_2(t+1)$ بسیار به معادله PSO بستگی دارد. مقدار تابع ملاک شایستگی $x_2(t+1)$ برابر با $f(x_2(t+1))=275$است، که حتی از $\hat{y}_i$ نیز بهتر است و باید مقدار $\hat{y}_i$ تغییر داده شود. اگر چه تابع ملاک شایستگی در $x_2(t)$ از تابع ملاک شایستگی $x_2(t+1)$ بهتر است اما در زمانt یکی از ابعاد به جواب بهینه 20 بسیار نزدیک است. که در مرحله t+1 به ۵ رسیده است. در واقع اینجا اطلاعات مفید از دست داده شده است. همین مشکل را می‌توان به ابعاد بزرگتر تعمیم داد. این رفتار دقیقا مانند رفتاری است که دو قدم به جلو برداشته شود و سپس یک قدم به عقب. این خطا دقیقا زمانی است که تمام ابعاد بردار به روز می‌شوند. یکی از نمونه الگوریتم هایی که سعی در بر طرف نمودن این مشکل نموده است، CPSO نام دارد

[1] Global minima

۱-۲-۳-۲-۲- انواع پایه ای از PSO

۱.۲.۳.۲.۱ مهار کردن سرعت

نحوه جستجوی عمومی و جستجوی محلی یک الگوریتم عوامل مهمی جهت ارزیابی یک الگوریتم می باشند. جستجوی عمومی یعنی اینکه الگوریتم به چه میزان قادر است ناحیه‌های متفاوتی از فضای جستجو را اکتشاف کند. در مقابل جستجوی محلی بدین مفهوم است که الگوریتم فقط ناحیه‌هایی را که احتمال وجود جواب در آنها وجود دارد، جستجو نماید. الگوریتم بهینه سازیی خوب است که تعادلی را بین این دو گونه جستجو بر قرار نماید. در بیشتر الگوریتمها برای ایجاد تعادل بین جستجوی عمومی و محلی معادله سرعت ذره را تغییر می‌دهند و سرعت ذره گاهی بسیار زیاد می‌شود. برای مهار نمودن سرعت ذره در[۹] یک حد آستانه ای به نام V_{max} برای سرعت ذره در نظر گرفته شده است که سرعت ذره نباید از این حد فراتر برود.

$$v_{i,j}(t+1) = \begin{cases} v'_{i,j}(t+1) & if \;\; v'_{i,j}(t+1) < V_{\max,j} \\ V_{\max,j} & if \;\; v'_{i,j}(t+1) \geq V_{\max,j} \end{cases} \qquad (۹)$$
$$V_{\max,j} = \delta(x_{\max,j} - x_{\min,j})$$

۱.۲.۳.۲.۲ وزن اینرسی

وزن اینرسی ضریب سرعت قبلی ذره می‌باشد که مشخص می‌نماید سرعت قبلی ذره تا چه اندازه در محاسبه سرعت فعلی ذره دخیل است. جهت تنظیم این پارامتر روشهای زیادی ارائه شده است که در ادامه به آنها می پردازیم:

۱.۲.۳.۲.۲.۱ مقدار تصادفی

وزن اینرسی را می‌توان به طور تصادفی مقداردهی نمود. یک نوع مقدار دهی استفاده از توزیع گوسین می‌باشد $w \sim N(0.72, \sigma)$ است که σ را به گونه ای انتخاب می‌نمایند که w از ۱ بزرگتر نباشد. در[۱۰] وزن اینرسی را مقیاسی از مولفه های شناختی و اجتماعی در نظر ،گرفته است. $W=(c_1r_1+c_2r_2)$

۱.۲.۳.۲.۲.۲ کاهش خطی وزن اینرسی

وزن اینرسی را در ابتدای الگوریتم مقدار زیادی (معمولا ۰.۹) در نظر گرفته و سپس به مرور زمان آنرا کاهش می دهند (معمولا ۰.۴). معادله این تغییر خطی در رابطه (۱۰)نمایش داده شده است:

$$w(t) = \left(w(0) - w(n_t)\right)\frac{(n_t - t)}{n_t} + w(n_t) \qquad (۱۰)$$

Nt ماکزیمم تعداد تکرارهای الگوریتم می‌باشد. w(0) وزن اینرسی اولیه است و w(nt) مقدار نهایی وزن اینرسی می‌باشد. و w(t) وزن اینرسی در تکرار t ام الگوریتم می‌باشد. در [۱۰] وزن های اینرسی از ۰.۴ به ۰.۹ افزایش داده شده است.

۱.۲.۳.۲.۲.۳ کاهش غیر خطی وزن اینرسی

در این الگوریتم وزن اینرسی به طور غیر خطی کاهش می‌یابد این امر باعث می‌شود که ذره در فضای جستجو بیشتر جستجوی عمومی داشته باشد. در [۱۱] از رابطه (۱۱)جهت کاهش غیر خطی وزن اینرسی شده است.

$$w(t+1) = \frac{\left(w(t) - 0.4\right)(n_t - t)}{n_t + 0.4}, w(0) = 0.9 \qquad (۱۱)$$

در [۱۲] از رابطه (۱۲)جهت کاهش غیر خطی وزن اینرسی استفاده شده است

$$w(t+1) = \alpha\ w(t') \quad .\alpha = 0.975 \qquad (۱۲)$$

در[۱۳] روش دیگری که جهت تنظیم وزن استفاده شده در رابطه (۱۳)آورده شده است:

$$w_i(t+1) = w(0) + \left(w(n_t) - w(0)\right)\frac{e^{m_i(t)} - 1}{e^{m_i(t)} + 1}$$
$$m_i(t) = \frac{f(\hat{y}_i(t)) - f(x_i(t))}{f(\hat{y}_i(t)) + f(x_i(t))} \qquad (۱۳)$$

که w(nt)تقریبا برابر با ۰.۵ است و w(0)<1 می باشد.

۱.۲.۳.۲.۲.۴ وزن اینرسی تطبیقی فازی

در این روش وزن اینرسی به طور اتوماتیک توسط مجموعه و قوانین فازی تنظیم میگردد[۱۴]. سیستم فازی که برای این کار در نظر گرفته شده است، به صورت زیر می باشد:

ورودی های سیستم فازی وزن اینرسی ذره و بهترین موقعیت ملاقات شده توسط کل گروه می باشند. خروجی سیستم فازی مقدار وزن اینرسی جدید ذره می باشد. سه مجموعه فازی به نامهای LOW, HIGH,MEDIUMوجود دارند که توابع عضویت این مجموعه ها مثلثی شکل در نظر گرفته شده اند. تعداد قوانین فازی شش قانون است که نمونه ای از قوانین در زیر آورده شده است

If normalized best fitness isLOW ,and current inertia weight value is LOW then the change in weight is Medium

۱.۲.۳.۲.۲.۵ فاکتور انقباض

از یک فاکتور انقباض برای اطمینان از همگرا شدن استفاده شده است. از این فاکتور برای انتخاب مقادیر w، c_1، c_2 استفاده می گردد.معادله بهنگام شدن سرعت در رابطه (۱۴)آورده شده است. [۱۵]

$$v_{i,j}(t+1) = \chi(v_{i,j}(t) + c_1 r_{1,j}(t)(y_{i,j}(t) - x_{i,j}(t)) + c_2 r_{2,j}(t)(\hat{y}_j(t) - x_{i,} \quad (۱۴)$$

χ : یک فاکتور انقباض است که به صورت(۱۵)تعریف می‌گردد.

$$\chi = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}}, \quad \phi = c_1 + c_2, \phi > 4 \quad (۱۵)$$

استفاده از فاکتور انقباض و تغییر بردار سرعت کارایی و نرخ همگرا شدن را افزایش می‌دهد.

۱.۲.۳.۲.۳ به روز کردن همگام[1] و نا همگام[2]

در الگوریتمهای gbest PSO و lbest PSO می توان معادله سرعت ذرات را به طور همزمان و یا ناهمزمان به روز نمود. اگر در ابتدا هر ذره موقعیتش و بهترین موقعیت ملاقات شده‌اش، به روز گردد و در پی آن بهترین موقعیت ملاقات شده توسط کل گروه به روز گردد، به این حالت به روز نمودن غیر همگام گفته می شود و اگر تمام ذرات به طور همزمان موقعیت و بهترین موقعیت ملاقات شده خود را به روز نمایند و سپس بهترین موقعیت ملاقات شده کل گروه به روز شود به این حالت ، به روز کردن همزمان یا همگام گفته می شود. به روز نمودن غیر همگام این حسن را دارد که ذرات خیلی سریع از ناحیه های خوب در زمان اجرا آگاه می‌شوند، برعکس در روش همگام ذرات در هر تکرار از ناحیه‌های خوب آگاه شده و از محیط بازخورد می‌گیرند. به روز نمودن همگام برای الگوریتم gebst PSO مناسب است و به روز نمودن غیر همگام برای الگوریتم lbestPSO مناسب است. [۱۱]

۱.۲.۳.۲.۴ پارامترهای PSO استاندارد

پارامترهای زیادی هستند که بر روی الگوریتم PSO تاثیر می گذارند. به عنوان مثال تعداد ذرات، ابعاد مساله، وزن اینرسی، اندازه همسایه‌ها، تعداد تکرارهای الگوریتم، ضرایب ثابت مولفه‌های اجتماعی و مولفه‌های شناختی و ماکزیمم سرعت یک ذره.

[1] Synchronous
[2] Asynchoronous

۱.۲.۳.۲.۴.۱ اندازه گروه

هر گاه تعداد ذرات زیاد باشد و ذرات در فضای جستجو به طور یکنواخت توزیع شده باشند، تنوع در بین ذرات زیاد شده و الگوریتم راندمان بالاتری می یابد. البته باید توجه شود که تعداد زیاد ذرات در پیچیدگی الگوریتم ارتباط مستقیم دارند. هر چند که نسبت زمانیکه تعداد ذرات کم است، تعداد تکرارهای الگوریتم کمتر است و زمان رسیدن به پاسخ بهینه کمتر است[۱۶]. مقدار مناسب ذرات به مساله بستگی دارد.

۱.۲.۳.۲.۴.۲ اندازه همسایگی

اندازه همسایه تاثیر زیادی در تعامل بین ذرات دارد. سایز همسایگی کم ، تعامل بین ذرات را کاهش می‌دهد و نرخ همگرایی در این روش کم بوده و امکان قرار گرفتن ذره در مینیمم محلی را کاهش می دهد. جهت استفاده از فواید همسایگی بزرگ و کوچک، در ابتدا اندازه همسایگی را کوچک و سپس بزرگ در نظر گرفته می شود[۱۷].

۱.۲.۳.۲.۴.۳ ضرایب سرعت۱

c_1,c_2 با ضرایب r_1,r_2 در مولفه‌های اجتماعی و شناختی سرعت ذره نقش بسیار زیادی در راندمان ذره دارند. اگر $c_1,c_2=0$ باشد ذرات فقط با سرعت خاصی بدون هدف در فضا حرکت می نمایند و آنقدر حرکت می نماید تا به مرز فضای جستجو برسد. اگر $c_1>0$, $c_2=0$ باشد، ذرات فقط به تجربه فردی خود توجه می نماید. اگر $c_2>0$, $c_1=0$ باشد، ذرات فقط به بهترین فرد گروه توجه می‌نماید. معمولا در بسیاری از الگوریتمها c_1,c_2 را ثابت در نظر میگیرند. در[۱۳] این پارامترها در طول الگوریتم یاد گرفته می شوند. به عنوان مثال درمعادله(۱۶)تنظیم پارامتر c_2 نشان داده شده است:

$$c_2(t)=\frac{c_{2,\min}+c_{2,\max}}{2}+\frac{c_{2,\max}-c_{2,\min}}{2}+\frac{e^{-m_i(t)}-1}{e^{-m_i(t)}+1}$$
$$m_i(t)=\frac{f(\hat{y}_i(t))-f(x_i(t))}{f(\hat{y}_i(t))+f(x_i(t))} \qquad (۱۶)$$

در[۱۱] پارامترهایc_1,c_2 به صورت خطی کاهش داده شدند که نتایج چندان جالبی را در بر نداشت. در [۱۸] سعی شده است که به صورت خطی پارامترها را تغییر دهند با این تفاوت که c_1 کاهش و c_2 افزایش داده شوند. این استراتژی بر روی جستجوی عمومی ذرات در فضای جستجو در ابتدای الگوریتم تاکید دارد. و در انتهای الگوریتم به جستجوی محلی ذرات اهمیت داده و ذرات به سمت بهترین موقعیت ملاقات شده گروه جذب می شوند. پارامترها به صورت معادله(۱۷)به روز می گردند.

---

[1] Acceleration Coefficients

$$c_2(t) = (c_{2,\max} - c_{2,\min})\frac{t}{n_t} + c_{2,\max}$$

$$c_1(t) = (c_{1,\max} - c_{1,\min})\frac{t}{n_t} + c_{1,\max} \qquad (۱۷)$$

where $c_{1,\max} = c_{2,\max} = 2.5$ *and* $c_{1,\min} = c_{2,\min} =$

در [۱۳] جهت تنظیم پارامترهای c1,c2 از سیستم فازی استفاده شده است. ورودی‌های سیستم فازی تنوع ذرات و تعداد تکرار جاری الگوریتم می‌باشند. خروجی های سیستم فازی نیز c_1,c_2 هستند. توابع عضویت فازی را می‌توان توابع MEDIUM, LOW,HIGH در نظر گرفت. قوانین و مجموعه های فازی در جدول ۱۰- نمایش داده شده اند.

جدول ۱۰-): قوانین فازی برای تنظیم پارامترهای c1,c2

RuleNumber	Inputs			Outputs
	Diversity(S,t)		1	C2
1	L			H
2	L			M
3	L			L
4	M			M
5	M			M
6	M			L
7	H			H
8	H			M
9	H			L

۱-۳- انواع روشهای بهینه سازی گروه ذرات

الگوریتمهای PSO در حالت کلی به دسته های زیر تقسیم می شوند:

بهینه سازی گروه ذرات تک هدفی[۲۵]: دسته الگوریتم هایی که تنها پاسخ یک مسئله بهینه سازی پیوسته و بدون قید را پیدا می کنند.

Niching PSO : الگوربتم هایی را شامل می شود که توانایی تعیین بیش از یک جواب مسئله بهینه سازی را دارند. الگوریتم‌های Niching مدل دیگری از پردازش های طبیعی می‌باشند که عده‌ای از افراد برای یافتن منابع محدود در طبیعت با یکدیگر رقابت می‌کنند.در [۱۶] [۱۹] از الگوریتم PSO برای یافتن نقاط بهینه یک تابع استفاده شده است.

بهینه سازی گروه ذرات مقید ومحدود شده۲۶: الگوریتم هایی از بهینه‌سازی گروه ذرات را شامل می‌شوند که برای حل مسائل دارای قید توسعه داده شده اند. بسیاری از مسائل مهندسی مسائلی دارای محدودیت می‌باشند.

بهینه سازی گروه Multi Objective۲۷: الگوریتم‌هایی از بهینه‌سازی گروه ذرات را شامل می‌شوند که برای مسائل با چندین زیر هدف متناقض بکاربرده می شوند[۲۰].

بهینه سازی گروه با محیط متغییر۲۸: الگوریتم هایی از بهینه سازی گروه ذرات را شامل می شوند که قادرند نقاط بهینه را در فضای جستجوی پویا پیدا نموده و دنبال نمایند

بهینه سازی گروه ذرات در فضای گسسته۲۹: الگوریتم هایی از بهینه سازی گروه ذرات را شامل می شوند که برای مسائل با بهینه سازی در فضای گسسته بکاربرده می شوند.

۱-۳-۱- بهینه سازی گروه ذرات تک هدفی[۳۰]

[25] Single –solution PSO (single Objective)
[26] Constrained PSO
[27] Multi Objective PSO
[28] Dynamic environment PSO
[29] Discrete PSO
[30] Single –solution PSO (single Objective)

بعد از اینکه مدل اصلی PSO ارائه شد، تحقیقات بسیاری جهت از بین بردن نقاط ضعف PSO صورت گرفت که در زیر به نمونه هایی از این روشها می پردازیم. در PSO همانند الگوریتم های ژنتیک تضمینی برای یافتن بهترین پاسخ وجود ندارد و امکان دارد که این الگوریتم در مینیمم‌های محلی قرار بگیرد و به همگرایی زودرس برسد. برای رفع این مشکلات روشهای زیادی ارائه شده است. الگوریتم های این دسته را می توان به ۵ دسته زیر تقسیم نمود:

الگوریتم های مبتنی بر اجتماع: این دسته شامل الگوریتم‌هایی هستند که از توپولوژی‌های اجتماعی متفاوتی برای همسایگی ذرات استفاده می‌کنند. و همچنین در این گونه از الگوریتم‌ها مدلهای مختلفی برای ردو بدل اطلاعات در بین ذرات وجود دارد.

الگوریتم های ترکیبی: این دسته شامل الگوریتم های ترکیبی از PSO با الگوریتم های دیگر می باشند.

الگوریتم های چند جمعیتی : این دسته از الگوریتم‌ها به جای استفاده از بیش از یک دسته گروه ذرات استفاده می‌نمانید. این گروه‌ها می‌توانند کاملا مجزا با یکدیگر رفتار نمایند یا می توانند با یکدیگر نیز در تعامل باشند.

الگوریتم بر پایه فرهنگ[۳۱]: در این روشها مکانیزم جستجوی محلی با الگوریتم PSO استاندارد ادغام شده است.

چند نقطه شروع:در این نوع الگوریتم ها هنگامیکه الگوریتم به مرحله خاصی از جستجو می رسد ذرات کل گروه یا قسمتی از گروه را دوباره در فضای جستجو قرار می دهند

الگوریتم های جذبی[۳۲]: در این گونه از الگوریتم‌ها سعی میشود که از برخورد ذرات جلوگیری شود و یا ذرات جذب یکدیگر شوند.هدف از این دسته الگوریتم‌ها افزایش تنوع در بین ذرات می‌باشد.

۱-۳-۱-۱- الگوریتم های مبتنی بر اجتماع

گونه‌های متفاوتی از توپولوژی های شبکه برای همسایگی ذرات در نظر گرفته شده است. که از این نوع توپولوژی ها می توان به شبکه های ستاره ای، وان نیومن و حلقه ای اشاره نمود. الگوریتمهای دیگری نیز وجود دارند که براساس تقسیم اطلاعات در بین ذرات بنا نهاده شده‌اند. استراتژیهای متفاوتی وجود دارد که

[31] Memetic algorithm
[32] Repelling methods

اطلاعات در بین ذرات به اشتراک گذاشته شوند. و ذرات همسایگی خود را تغییر دهند.ساختارهای متفاوتی از شبکه‌های اجتماعی وجود دارد.

۱.۳.۱.۱.۱همسایگی قابل رشد

در شبکه های اجتماعی هر اندازه که تعامل بین ذرات کم باشد، ذرات دیرتر همگرا می‌شوند و به ذرات اجازه داده می شود که بیشتر در فضا جستجو انجام دهند. توپولوژی هایی که ذرات کاملا با یکدیگر در تعامل باشند همگرایی بسیار سریع رخ میدهد. جهت استفاده از دو مزیت همگرایی سریع و جستجوی فضا، در [۱۷] این دو نمونه از همسایگی با یکدیگر ترکیب شده‌اند. در ابتدا همانند الگوریتم lbest PSO سایز همسایگی را ۲ قرار داده و سپس در طی تکرار های الگوریتم سایز همسایگی افزایش یافته تا تمام ذرات را در بر بگیرد.

۱.۳.۱.۱.۲ شبکه های اجتماعی دنیای کوچک[۳۳]

در [۲۱] نوع خاصی از شبکه‌ها ارائه شده است که ذرات به صورت تصادفی به یکدیگر متصل می‌گردند. این اتصال تصادفی ذرات به یکدیگر دنیای کوچک نامیده می‌شود. که در آن شبکه حلقه‌ای با بردارهای n_s تایی تعریف شده که به ازای هر بردار n_N یک اتصال وجود دارد. برای هر اتصال به طور تصادفی یک احتمال در نظر گرفته می‌شود. اتصالی که احتمال آن یک است در نظر گرفته می‌شود.

۱.۳.۱.۱.۳ شبکه‌های اجتماعی سلسله مراتبی[۳۴]

در [۲۳] یک شبکه اجتماعی سلسله مراتبی ارائه شده است که ذرات در درون یک درخت قرار داده می شوند. بهترین ذرات در ریشه درخت قرار دارند و ذرات بد در برگها واقع شده‌اند. در سطوح میانی درخت، ذراتی قرار دارند که نسبت به فرزندان خود بهتر می‌باشند. طی جستجو اگر فرزندان نسبت به والدین خود موقعیت بهتری را ملاقات نمایند ،آنگاه فرزند جایش را با والد خود عوض می نماید. با این کار ذرات با موقعیت بهتر همیشه در سطوح بالاتر وجود دارند. $\vee H-PSO$ و $\wedge H-PSO$ دو نمونه از این الگوریتم ها می باشند.

[33] Small World Social Network
[34] Hierarchical Social Network

۱.۳.۱.۱.۴ شبکه های اجتماعی فضایی

در بیشتر الگوریتم‌ها همسایه‌های یک ذره بر اساس شماره اندیس ذره در نظر گرفته می‌شوند. اما در [۲۴] همسایه یک ذره از نظر فاصله اقلیدسی ذرات در فضای جستجو مشخص شده است. محاسبه فاصله اقلیدسی ذرات زمان زیادی از اجرای الگوریتم را به خود اختصاص می‌دهد. در رابطه (۱۸)محاسبه همسایگی در فضا ارائه شده است:

$$
\begin{aligned}
&Calculate\ the\ EuclidanceDis\tan ce \quad \varepsilon(x_{i1}\ , x_{i2}) \quad \forall i1, i2 = 1,....,n_{si} \\
&S = \{i : i = 1,...n_s\} \\
&for\ i = 1,....,n_s \quad do \\
&\quad S^{'} = S_i \\
&\quad for \quad i^{'} = 1,....,n_{Ni'} \quad do \\
&\qquad N_i = N_i \cup \{x_{i''} \ : \varepsilon(x_{i'}\ , x_{i''}) = \min\{\varepsilon(x_i\ , x_{i'''}),\ \forall_{x_{i'''}} \in S'\}; \\
&\qquad S'' = S''\backslash\{x_{i''}\}; \\
&\quad end \\
&end
\end{aligned}
\qquad (۱۸)
$$

۱-۳-۱-۲- استراتژی‌های اشتراک اطلاعات

تا بحال هر گاه صحبت از همسایگی به میان می‌آمد در پی آن گفته می‌شد که ذرات بهترین موقعیت ملاقات شده در بین همسایگان و یاگروه را جهت به روز نمودن سرعت خود استفاده می‌نمایند. اما عوامل دیگری نیز وجود دارند که ذرات بخواهند به اشتراک بگذارند. در ادامه به بررسی این عوامل می پردازیم.

۱.۳.۱.۲.۱ مدل PSO کاملا آگاه Fully informed PSO

در[۲۵] الگوریتمی ارائه شده است که موقعیتهای تمام ذرات موجود همسایه در عملکرد ذره دخیل می باشند. این نوع خاص ، FIPSO نامیده می شود. دو گونه متفاوت از این الگوریتم ارائه شده است: در روش اول هر ذره دارای Ni ذره در همسایگی خود است که برای به روز نمودن سرعت ذره به جای استفاده ار مولفه های شناختی و اجتماعی از عبارت دیگری که در معادله(۱۹)دیده می شود، استفاده می‌نماید.

$$ {}_i^d(t+1) = \chi\left(v_i^d(t) + \sum_{m=1}^{nN_i} \frac{r(t)(y_m(t) - X_i(t)}{nN_i} \right) $$
$$ (t) \sim U(0, c_1 + c_2)^{nx} \qquad (۱۹) $$

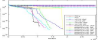
در روش دوم برای به روز نمودن سرعت ذره از معادله ۲۰استفاده شده است. یکی از معایب این الگوریتم‌ها در این است که ذرات بر روی یکدیگر تاثیر می گذراند. به عنوان مثال اگر سرعت یک ذره a و سرعت یک ذره دیگر a– باشد این دو سرعت بر روی یکدیگر تاثیر می گذارند و مجموع این دو عدد برابر با ۰ میگردد. برای رفع این مشکل سرعت ذرات را با یک وزن خاصی در محاسبات تاثیر می دهد(i(t.χ دو روش ارائه داده شده این الگوریتم راندمان بالاتری نسبت به PSO استاندارد دارند[۲۵].

$$ + 1) = \chi\left(v_i^d(t) + \frac{\sum_{m=1}^{nNi} \frac{\phi_1 P_m(t)}{f(x_m(t))}}{\sum_{m=1}^{nNi} \frac{\phi_m}{f(x_m(t))}} \right), \phi_m \sim U(0, \frac{c_1 + c_2}{nN_i})^{nx} $$
$$ t) = \frac{\phi_1 y_m(t) + \phi_1 \hat{y}_m(t)}{\phi_1 + \phi_2} \qquad (۲۰) $$

۱.۳.۱.۲.۲ مدل FDR-PSO[35]

اخیراً در سال ۲۰۰۳ راه حل اصلی برای PSO ارائه شد که یک عبارت جدید به معادله بردار سرعت اضافه نموده است. این عبارت جدید اجازه می‌دهد که هر ذره به سمت همسایه‌ای که "بهترین موقعیت شخصی " بهتری دارد حرکت کند. معادله بردار سرعت به صورت زیر می باشد.

$$ +1) = wv_{1,j}(t) + \psi_1(y_{i,j}(t) - x_{i,j}(t)) + \psi_2(\hat{y}_j(t) - x_{i,j}(t)) + \psi_3(y_{\eta,j}(t) - x_{i,j}(t)) \qquad (۲۱) $$

ψ1, ψ2, ψ3 پارامترهایی هستند که توسط کاربر مشخص می‌گردند و هر  توسط معادله(۲۲)بیشینه می گردد:

[35] Fitness-Distance Ratio based PSO

$$\frac{f(x_i(t)) - f(y_\eta(t))}{|\, y_\eta(t) - x_{i,j}(t)\,|} \qquad (۲۲)$$

FDR-PSO احتمال همگرایی زودرس را کاهش می‌دهد. بنابراین احتمال اینکه در کمینه محلی بیافتد کاهش می‌یابد. به علاوه اینکه FDR-PSO که دارای پارامترهای زیر باشد $\psi1=\psi2=1,\psi3=2$ کارایی بالاتری از PSO و ARPSO وDPSO وSOC-PSO وMulti Swarm PSO دارد .

۱.۳.۱.۲.۳ مدل barebones PSO

درمعادله سرعت در الگوریتم PSO استاندارد به یک نسبتی مولفه های اجتماعی و مولفه های شناختی را در معادله سرعت خود دخیل می کردند این نسبت با c_1,c_2 مشخص می‌شد. اگر $c_1=c_2$ بنابراین ذرات در هر بعد خود به $\left(y_{i,j}(t)+\hat{y}_{i,j}(t)\right)/2$ همگرا می‌شوند. در این الگوریتم سرعت ذره را به صورت معادله(۲۳)به روز می شود.

$$\begin{aligned} v_{i,j}(t) &\sim N\left(\frac{y_{i,j}(t) + y_{i,j}(t)}{2}, \sigma\right) \\ \sigma &= |\, y_{i,j}(t) + \hat{y}_{i,j}(t)\,| \\ x_{i,j}(t+1) &= v_{i,j}(t+1) \end{aligned} \qquad (۲۳)$$

مدل دیگری از معادله سرعت در این الگوریتم نیز به صورت معادله(۲۴)می باشد

$$v_{i,j}(t+1) = \begin{cases} y_{i,j}(t) & if \;\; U(0,1) < 0.5 \\ N\left(\frac{y_{i,j}(t) + \hat{y}_{i,j}(t)}{2}, \sigma\right) & otherwise \end{cases} \qquad (۲۴)$$

۱.۳.۱.۲.۴ استراتژی اشتراک اطلاعات عمومی گروه

در [۲۶] از استراتژی اشتراک اطلاعات عمومی گروه استفاده شده است. در این روش ذرات به سمت یک راهنما کشیده می‌شوند که این راهنما بهترین راندمان را در کل گروه دارد بر خلاف PSOاستاندارد که از بین همسایگی ذرات بهترین ذره را انتخاب می نماید، در این روش بهترین ذره از بین انبوهی از ذرات که از میانگین گروه بهتر عمل نموده اند، انتخاب می گردد.

۱.۳.۱.۲.۵ مدل UPSO[36]

راندمان الگوریتم PSO بستگی زیادی به جستجو های محلی و عمومی کل گروه دارد .همیشه باید بین این دو یک تعادل وجود داشته باشد. در صورتیکه یک ذره در زمان یه روز کردن موقعیت و سرعت خود کل گروه را در نظر بگیرد-global varient- و بهترین مقدار گروه را در معادله سرعت خود قرار دهد، ذرات به سرعت به سمت بهترین ذره جمع شده و همگرا می‌شوند، بر عکس اگر یک ذره در زمان به روز کردن موقعیت و سرعت خود تعداد محدودی از ذرات گروه را در نظر بگیرد-local varient- همگرایی زود رس ذرات کاهش یافته و از قرار گرفتن و مینیمم محلی نیز جلو گیری می شود. در نتیجه جستجوی محلی در بین ذرات فابلیت جستجوی عمومی بین ذرات را بالا می‌برد. انتخاب مناسب تعداد همسایه های یک ذره یکی از مسائلی است که تحقیق بر روی آن کماکان ادامه دارد. UPSO یکی از روشهایی است که قابلیت های جستجوی عمومی و محلی را با یکدیگر ترکیب می‌نماید.

 : معادله به روز شدن سرعت ذره xi است که همسایه‌های آن کل ذرات گروه می باشند

 :معادله به روز شدن سرعت ذره xi است که همسایه های آن تعداد محدودی از ذرات گروه می باشند

$$g_i^{(k+1)} = \chi\left[v_i^{(k)} + c_1 r_1 \left(p_i^{(k)} - x_i^{(k)}\right) + c_2 r_1 \left(p_g^{(k)} - x_i^{(k)}\right)\right]$$
$$L_i^{(k+1)} = \chi\left[v_i^{(k)} + c_1 r_1' \left(p_i^{(k)} - x_i^{(k)}\right) + c_2 r_2' \left(p_g^{(k)} - x_i^{(k)}\right)\right] \quad (۲۵)$$

Kشماره تکرار الگوریتم را نشان می‌دهد ، g اندیش بهترین ذره در کل گروه می باشد(global variant) و g_i اندیس بهترین ذره در بین همسایگان ذره x_i می باشد(local variant). در مدل UPSO معادله به روز نمودن سرعت که ترکیبی از $g_i^{(k+1)}$ و $L_i^{(k+1)}$ است به صورت زیر می باشد:

$$U_i^{(k+1)} = u g_i^{(k+1)} + (1-u) L_i^{(k+1)}$$
$$x_i^{(k+1)} = x_i^{(k)} + U_i^{(k+1)} \quad (۲۶)$$

$U \in [0,1]$ پارامتری است که فاکتور یکسان سازی[37] نامیده می شود. و تاثیر دو قسمت local variant و global variant را در الگوریتم مشخص می تمایند. می توان مقدار u را به گونه ای انتخاب نمود که هم جستجوی عمومی و هم جستجوی محلی تاثیر داشته باشند و یا با u=0 تاثیر local variant را افزایش داد و

[36] Unified Particle Swarm Optimization
[37] Unification factor

یا با u=1 تاثیر global varient را افزایش داد. می توان در معادله(۲۷)از یک پارامتر تصادفیr_3 استفاده نمود و روش مذکور را بهبود بخشید.

$$U_i^{(k+1)} = r_3 ug_i^{(k+1)} + (1-u) L_i^{(k+1)}$$
$$U_i^{(k+1)} = ug_i^{(k+1)} + r_3 (1-u) L_i^{(k+1)} \quad (۲۷)$$

همانطور که از معادله بالا متوجه می شود می‌توان این پارامتر را در قسمت مربوط به local variant و یا قسمت مربوط به global variant اضافه نمود.$r3 \sim N(\mu,\sigma^2 I)$ پارامتر توزیع نرمال می‌باشد که I ماتریس با قطر یک می‌باشد. این روش نسبت به الگوریتم استاندارد PSO راندمان بالاتری دارد.

۱-۳-۱-۳- الگوریتم های ترکیبی

این دسته از الگوریتم ها ترکیبی از الگوریتم‌های بهینه سازی گروه ذرات و سایر الگوریتم ها چون الگوریتم ژنتیک، استراتژیهای تکاملی، تفاضل تکاملی، کلونی مورچه‌ها و سیستم های فازی می باشند در ادامه به آنها می پردازیم:

۱.۳.۱.۳.۱ الگوریتم PSO مبتنی بر الگوریتم ژنتیک

PSO یک الگوریتم پیوسته توارثی می‌باشد در صورتیکه الگوریتم ژنتیک[۳۸]GA یک الگوریتم گسسته توارثی است. با توجه به آزمایشاتی که صورت گرفته، مشخص شده است که در بعضی از مسائل بهینه‌سازی پیوسته PSO بهتر از GA عمل می‌نماید. همینطور PSO دودویی با الگوریتم ژنتیک مقایسه شده است و مشخص شده است که PSO دود,یی بسیار سریعتر، مقاومتر و کارایی بالاتری نسبت به الگوریتم ژنتیک دودویی دارد، مخصوصاً در مواردی که بعد مسائل زیاد است. روشهای ترکیبی بسیار زیادی از این دو گونه الگوریتم ارائه شده است. این روشها بدین صورت عمل می نمایند که بدون استفاده از تابع ملاک شایستگی جمعیت تولید می‌نماید و این جمعیت به عنوان جمعیت اولیه در الگوریتم دیگری بکار می‌رود. PSO_GA و GA_PSO دو نمونه از این الگوریتم ها می‌باشند.

در GA_PSO: از جمعیتهای الگوریتم ژنتیک برای مقدار دهی اولیه جمعیتهای PSO استفاده می‌گردد.

درPSO_GA : از جمعیتهای PSO برای مقدار دهی اولیه جمعیتهای الگوریتم ژنتیک استفاده می‌گردد.

[38] Genetic Algorithm

آزمایشات نشان داده که PSO_GA بهتر از PSO عمل می‌نماید و PSO وPSO_GA بهتر از روشهای GA و GA_PSO عمل می نمایند. یکی از ابتدایی ترین کاربردهای PSO استفاده این روش در آموزش شبکه‌های عصبی می‌باشد . نتایج نشان داده که الگوریتم های آموزشی PSO بهتر از GA عمل می نماید. کارایی PSO بستگی به سایز جمعیت دارد.(سایز جمعیت نباید بسیار کوچک باشد) PSO با سایز گروه کم مانند یک الگوریتم ژنتیک با سایز جمعیت زیاد عمل می نماید. در بعضی از کاربردها از PSO به عنوان آموزش وزنهای شبکه عصبی استفاده شده است که نسبت به استفاده از الگوریتم ژنتیک برای این کار، کارایی بالاتری دارد.

۱.۳.۱.۳.۱.۱ بهینه سازی گروه ذرات و روش انتخابی [۳۹]

در سال ۱۹۹۸ یک روش ترکیبی از PSO با متد انتخابی تورنمنت ارائه شد. هر ذره بر اساس کارایی‌اش در مقابل بعضی از اعضای گروه که به صورت تصادفی انتخاب شده‌اند رتبه بندی می‌گردد. بدین منظور به هر ذره‌ای که نسبت به رقبایش دارای fitness بهتری می‌باشد یک امتیاز مثبت داده می‌شود. سپس جمعیت را بر اساس امتیازهایشان مرتب سازی نزولی می‌نماید. نیمه پایینی جمعیت را با نیمه بالایی جمعیت جایگزین می‌نماید این مرحله باعث کاهش تنوع جمعیت میگردد. نتایج نشان داده که کارایی این روش بیشتر از الگوریتم PSO (بدون w و χ) که دارای یک می‌نیمم میباشد،است. اما برای توابعی که دارای چندین کمینه محلی می‌باشد این روش کارایی کمتری نسبت به PSO دارد. پس می توان نتیجه گرفت که هرچند این روش انتخابی قابلیت جستجوی محلی را افزایش می دهد اما قابلیت جستجوی عمومی را کاهش می‌دهد. بنابراین استفاده از این روش باعث همگرایی زود رس می‌گردد.

۱.۳.۱.۳.۱.۲ جهش[۴۰] و PSO

در سال ۲۰۰۳ یک روش ترکیبی از PSO وجهش گوسین پیشنهاد شد و همینطور یک روش ترکیبی از lbest&gbest PSO نیز ارائه گردید[۲۷] با توجه به نتایج روش lbest PSO با استفاده از متغیر اپراتور جهش غیر یکنواخت کارایی بالاتری نسبت به PSO وGCPSO دارد.

[39] selection
[40] Mutation

۱.۳.۱.۳.۱.۳ تولید مجدد ذرات[۴۱]

در این دسته از الگوریتم ها ذرات موجود در گروه را دوباره ایجاد می‌نمایند. روشهای مختلف ایجاد ذرات وجود دارد که در Cheap PSO نمونه هایی از آنها آورده شده است. به عنوان مثال برای گروه ذرات جدید ایجاد شوند، ذرات خودشان را نابود کنند و یا اینرسی یا سرعت ذرات و پارامترهای c_1,c_2 ذرات تغییر داده شود

۱.۳.۱.۳.۲ بهینه سازی گروه ذرات مبتنی بر برنامه نویسی تکاملی

برنامه‌نویسی تکاملی گونه‌ای از الگوریتم های تکاملی است که مکانیزم تولید آن فقط بر اساس جهش می‌باشد. گونه‌های متفاوتی از جهش در برنامه‌نویسی تکاملی بکار رفته است که می‌توان از آن جمله به جهش گوسین[۲۸] ، جهش کوشی [۲۹]، جهش خطی [۳۰] و غیر خطی[۳۱] اشاره نمود. یا ترکیب این روش و PSO استاندارد الگوریتم‌هایی با راندمان بالایی نسبت به PSO استاندارد ایجاد شده‌اند. جهش می‌تواند در قسمتهایی از الگوریتم PSO استفاده گردد.

۱.۳.۱.۳.۳ بهینه سازی گروه ذرات مبتنی بر استراتژی های تکاملی

در[۳۲] روشهایی از ترکیب بهینه سازی گروه ذرات و استراتژیهای تکاملی ارائه شده است. این روشها راندمان بالایی نسبت به PSO استاندارد دارند.

۱.۳.۱.۳.۴ بهینه سازی گروه ذرات مبتنی بر تکامل تفاضلی۴۲DEPSO

جهت ایجاد تنوع در الگوریتم استاندارد PSO الگوریتم جدیدDPSO ایجاد شد که از یک اپراتور استفاده می‌نماید که با احتمالc_1 در ذره x_{id} یک جهش تصادفی ایجاد می نماید. جهش می تواند به صورت زنگوله‌ای شکل مانند توزیع گوسی باشد. معادله سرعت به صورت زیر می باشد:

[41] Reproduction
[42] Differential Evolution

$$v_{id} = w.v_{id} + c_1\ rand()\,(p_{id} - x_{id}) + c_1\ rand()\,(p_{gd} - x_{id})$$
$$x_{id} = x_{id} + v_{id} \qquad (۲۸)$$

تابع گوسینی که برای ایجاد جهش استفاده می شود دارای میانگین $(p_{id}+p_{gd})/2$ و واریانس $|p_{id}-p_{gd}|$ می باشد. البته درصورتیکه $|p_{id}-p_{gd}|$ مقدار بسیار ناچیزی باشد، جهش چندان مناسب به نظر نمی رسد. که در این روش نقطه اختصاصی $\ddot{y}_{i,j}$ با صورت زیر محاسبه می گردد:

$$if\ (\ r_1(t) < p_c\ OR\ \ j = k_d)\ \ then\ \ \ddot{y}_{i,j}(t) = \hat{y}_{,j}(t) + \frac{(y_{1,j}(t) - y_{2,j}(t)) + (y_{3,j}(t) - y_{4}}{2} \qquad (۲۹)$$

که, $k_d\ (t) \sim U(0,N_d)$, $r_1(t) \sim U(0,1)$ می باشند $y_1(t), y_2(t),$ و $y_3(t)$ به صورت تصادفی از بین بهترین موقعیت ذرات انتخاب می گردند. و سپس بنابراین معادله $y_i(t) = \ddot{y}_i(t)$ برقرار است اگر و تنها اگر $f(y_i(t)) > f(\ddot{y}_i(t))$ باشد. استفاده $y_i(t)$ به جای $x_i(t)$ از عدم سازماندهی گروه جلوگیری می‌نماید. الگوریتم DCPSO مانند الگوریتم PSO عمل می‌نماید با این تفاوت که در هر تکرار معادله بالا نیز بکار می‌رود. الگوریتم DEPSO از اپراتور جهش مربوط به الگوریتم تفاضلی DE استفاده می نماید.

IF (rand() < CR OR d==k) THEN $T_{id} = P_{gd} + \sigma_{2,d}$

Kیک عدد تصادفی در بازه [D.. 1] بوده و مشخص کننده یک بعد از فضای جستجو می باشد و CR مقدار ثابت بازترکیبی است با توجه به آزمایشاتی که صورت گرفته DEPSO در کل کارایی بالاتری نسبت به PSO، DE، GA، ES، DPSO، Fuzzy-adaptivePSO دارد.

۱.۳.۱.۳.۵ الگوریتم های ترکیبی فازی و بهینه سازی گروه ذرات

در این گونه از الگوریتم‌ها از سیستم فازی جهت تنظیم بعضی از پارامترهای الگوریتم PSO استفاده شده است. که در ادامه به آنها می پردازیم:

۱.۳.۱.۳.۵.۱ بهینه سازی گروه ذرات و Fuzzy Adaptive Turbulent PSO

هنگامیکه ذرات در مینیمم محلی قرار می گیرند به مرور زمان با تکرار قدمهای الگوریتم سرعت ذرات کاهش می‌یابد به گونه‌ای که ذرات توانایی جستجوی نقاط جدید را نداشته و در یک نقطه ثابت بی حرکت می مانند. در [۳۳] ایده جدیدی ارائه شده است تا هنگامیکه ذره‌ای به چنین حالتی رسید سرعت آن دوباره از ابتدا مقدار دهی گردد. معادله سرعت ذره در معادله (۳۰)نمایش داده شده است.

$$v_{id} = w.\hat{v} + c_1 r_1(x_{ij}^{\#}(t-1) - x_{ij}(t-1)) + c_2 r_2(x_j^{\#}(t-1) - x_{ij}(t-$$

$$\hat{v} = \begin{cases} v_{ij} & if \mid v_{ij} \mid \geq vc \\ u(-1,1)\, v_{\max} / \rho & if \mid v_{ij} \mid < vc \end{cases} \qquad (۳۰)$$

U(-1,1) یک عدد تصادفی است که با توزیع یکنواخت در بازه[1,-1] مقدار دهی می شود. ρ فاکتور مقیاس جهت کنترل دامنه نوسانات ذره تا vmax می‌باشد. vc مینیمم ترشلد سرعت ذره می‌باشد.

تغییرات موقعیت ذره بستگی بسیاری به دو پارامتر ρ و vc دارد. مقدار پایین vc تناوب نوسانات را کاهش می دهد و به احتمال زیاد باعث می شود ذره از مینیم محلی بیرون بیاید. اما مقدار زیاد vcباعث می‌شود که جستجوی سراسری در فضا افزایش یابد. مقدار ρ مستقیما دامنه نوسانات سرعت ذره را مشخص می نماید. جهت تنظیم اتوماتیک vc و ρ می توان از کنترلر فازی استفاده نمود . در این الگوریتم سیستم فازی دارای دو ورودی می باشد: CV سرعت جاری ذره Current Velocityو CBPE بهترین موجود در مساله می باشد. که این مقدار باید نرمال شده باشد. که در معادله فرمول(۳۱)نرمال شده آن آورده شده است.

$$NCBPE = \frac{CBPE - CBPE_{\min}}{CBPE_{\max} - CBPE_{\min}} \qquad (۳۱)$$

CBPEmin و CBPmax به ترتیب تخمینی از بهترین و بدترین راه حل می باشند. گاهی هیچ تخمینی نسبت به این مقادیر وجود ندار بنابراین در حین الگوریتم این مقادیر را از ۱ به ۰ به صورت خطی کاهش میدهند. سیستم فازی دارای دو خروجی ρ , Vck می‌باشد. Vck میزان تغییرات ترشلد سرعت را طبق فرمول(۳۲)کنترل می کند.

$$v_c = e - [10\ (1 + V_{ck})] \qquad (۳۲)$$

سیستم فازی به صورت زیر ایجاد شده است:

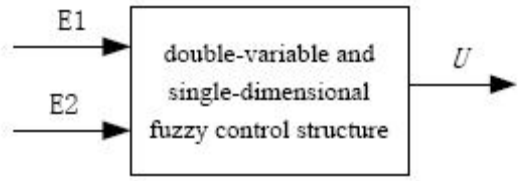
جدول ۰-۲) تنظیمات سیستم فازی و قوانین FTPSO

[System]	NumMFs=3
Name='*FAT PSO*'	MF1='Low':'trimf', [-1 -0.8 -0.5]
[Input1] Name='*NCBPE*'	MF2='Medium':'trimf', [-0.6 0 0.2]
Range=[0 1]	MF3='High':'trimf', [0.1 1.1 2.2]
NumMFs=3	[Output2]
MF1='Low':'gaussmf', [0.005 0]	Name='ρ'
MF2='Medium':'gaussmf', [0.03 0.1]	Range=[1 120]
MF3='High':'gaussmf', [0.25 1]	NumMFs=3
[Input2]	MF1='Small':'trimf', [1 1 4]

Name='*CV*'	MF2='Medium':'trimf', [2.214 10.71 59.29]
Range=[0 1e-006]	MF3='Large':'trimf', [47.15 120 120]
NumMFs=2	[Rules]
MF1='Low':'trapmf', [0 0 1e-030 1e-020]	1 1, 3 0 (1) : 1
MF2='High':'trapmf', [1e-010 1e-008 1e-006 1e-006]	2 0, 2 0 (1) : 1
	3 2, 1 0 (1) : 1
	1 1, 0 3 (1) : 2
	2 0, 0 2 (1) : 2
[Output1]	3 2, 0 1 (1) : 2
Name='*V ck*'	
Range=[-1 2.2]	

۱.۳.۱.۳.۵.۲ بهینه سازی گروه ذرات تطبیق داده شده با سیستم فازی Fuzzy Adaptive PSO

جستجوی PSO یک جستجوی غیر خطی و پویا می‌باشد. بنابراین هنگامیکه محیط به طور پویا تغییر می‌کند، پارامترهای ck نیز باید به صورت پویا در برنامه تغییر یابند. هر چند که PSO استاندارد توانایی تغییر این پارامترها را به صورت پویا ندارد. در FAPSO سعی شده که با استفاده از سیستم فازی تطبیقی این مقادیر نسبت به تغییرات محیط افزایش یا کاهش داد شوند. در شکل ۰-۷ ساختار این کنترلر فازی آورده شده است.



شکل ۰-۷):ساختار سیستم فازی با دو ورودی و یک خروجی

در این روش تاثیری که هر کدام از M موقعیت برتر بر روی یک ذره خاص دارند وابسته به فاصله ذره تا آن موقعیت برتر دارد. سیستم فازی این فاصله ها را به عنوان ورودی گرفته و میزان تاثیر این فاصله ها به عنوان خروجی به برنامه کاربردی داده می شود. $E_{k,id}= p_{k,id} -x_{id}$ فاصله ذره iام را نسبت به بهترین موقعیت ملاقات شده kام (در بعدd) مشخص می نماید.

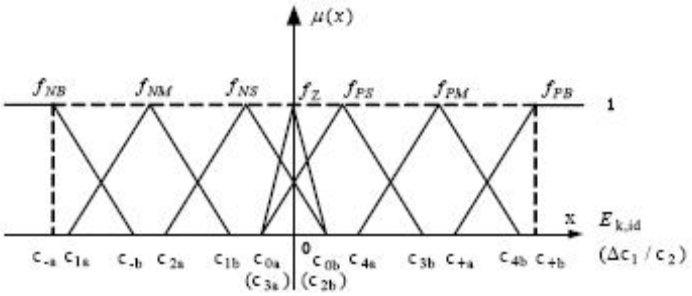
معادله سرعت ذره به صورت معادله(۳۳)می باشد

$$v_{id} = wv_{id} + \sum^{k} c_k rand_k () (P_{k,id} - x_{id}) \quad (۳۳)$$

M=2 و p1,id بهترین موقعیت ملا قات شده و $p_{k,id}$ دو موقعیت بهتر ملاقات شده توسط کل گروه می باشند.$E_{1,id}$ و $E_{2,id}$ به عنوان ورودی های سیستم فازی و c1/c2 به عنوان خروجی سیستم فازی می باشد. مجموعه های فازی به صورت زیر می باشند

(NB, NM, NS, Z, PS, PM, PB) که NB ، "Negative Big"است ، NM ، "Negative Medium"است، NS ، "Negative Small"است ، Z ، "Zero"است، PB ، "Positive Big" است ، PM ، "Positive Medium"است و PS ، "Positive Small"است.

در شکل زیر تابع عضویت فاری را مشاهده می نمایید:



شکل ۰-۸):توابع عضویت مجموعه فازی

۴۹ قوانین فازی نیز در جدول زیر آورده شده است

جدول ۰-۳): تنظیمات سیستم فازی

E_2 \ E_1	NB	NM	NS	Z	PS	PM	PB
NB	PS	PS	NM	NS	NS	Z	PS
NM	PM	PS	NS	NS	NS	Z	PS
NS	PB	PM	Z	NM	Z	PS	PM
Z	PB	PB	PM	Z	PM	PB	PB
PS	PB	PB	Z	NM	PS	PS	PB
PM	PM	PS	Z	NS	Z	PM	PB
PB	PS	Z	NS	NS	NS	PB	PB

الگوریتم فازی به صورت زیر می باشد:

ذرات را با توجه به الگوریتم MSPPSO مقدار دهی اولیه می نمایند.

تابع شایستکی هر ذره را ارزیابی می نمایند

Pbest تمام ذرات محاسبه شده و M ذره برتر مشخص می شوند

E1و E2 به عنوان ورودی به سیستم فازی داده میشوند

خروجی سیستم فازی $\Delta c1/c2$ محاسبه می گردد

مشخص می شود که ذره آیا به ناحیه بهتری رسیده یا نه . اگر نرسیده باشد سرعت بر طبق فرمول ازریابی می شود در غیر این صورت بر طبق فرمول PSO استاندارد ارزیابی می گردد.

تا زمانیکه الگوریتم به حداکثر تکرار نرسیده ادامه پیدا می کند.

۱.۳.۱.۳.۵.۳ PSO هوشمند:

پارامترهای w,c_1,c_2, نقش بسیار مهمی در رفتار همگرایی PSO ایفا می‌کنند. هر ذره جهت حرکت در فضای جستجو به یک میزان خاص به سمت این دو موقعیت gbest و pbest حرکت نموده و بر اساس این دو موقعیت، موقعیت و سرعت خود را به روز می‌نماید. اگر مقدار این پارامترها پایین باشد، به ذره کمک می‌کند تا با فاصله زیادی نسبت به سایر پاسخ بهینه حرکت نماید و در صورتیکه این پارامترها بالا باشند ذره به طور ناگهانی به سمت این دو نقطه مینیمم حرکت می‌نماید و دیگر فرصت جستجوی خود در محیط را از دست می‌دهد. در فرایند جستجو بعضی از مواقع رفتار اجتماعی مهمتر از رفتارهای شناختی ذرات است و بر عکس در مواردی دیگر رفتارهای شناختی خود ذره مناسب تر از رفتارهای اجتماعی ذرات است. میتوان این رفتارهای اجتماعی و شناختی ذره را با استفاده از یک سری از پارامترهای آماری تنظیم نمود که ذرات گروه در مسیر درستی حرکت نمایند. به عنوان مثال در ابتدای جستجوی ذرات در فضا، جستجوی سراسری مناسبتر از جستجوی محلی ذرات می باشد. در این مرحله می توان w و پارامتر رفتار اجتماعی ذرات را افزایش و پارامتر شناختی ذرات را کاهش داد. در این صورت ذرات می توانند حداکثر تعامل را با یکدیگر داشته باشند و ناحیه بهتر(با شایستگی بهتر) را تشخیص دهند. پس از یافتن ناحیه مناسب ذرات رفته رفته جستجوی محلی خود را آغاز می‌نمایند به گونه‌ای که پارامتر شناختی ذرات افزایش و پارامتر اجتماعی و W ذرات کاهش می‌یابد. همانطور که تابع شایستگی ذره بهتر و بهتر می‌شود باید ذره فضای جستجوی خود را کاهش دهد اما با دقت بیشتری درآن فضا حرکت نماید.این بدین مفهوم است که تعامل اجتماعی ذرات باید کاهش یابد. هنگامیکه ذرات به صورت محلی یک منطقه را جستجو می نمایند اگر به یکدیگر توجه نمایند و رفتار اجتماعی داشته باشند سریعاً همگرا می شوند چه بسا که این همگرایی زودرس باشد.

در بعضی از حالات که در عملکرد حرکت ذره بهبودی حاصل نشود ذره باید به اجتماع ذرات اطراف خود نگاه کند. بدین مفهوم که رفتار اجتماعی ذره افزایش و رفتار شناختی ذره کاهش داد شود. زمانیکه ذرات در مینیمم محلی قرار می گیرند بدین مفهوم است که gbest و pebst آن ذرات با یکدیگر برابر شده

است و در نتیجه به مرور زمان سرعت آن ذرات نیز کاهش می یابد به حدی که سرعت ذره به صفر می رسد. و دیگر با افزایش میزان w, c1,c2 امکان خارج شده از این وضعیت وجود ندارد. جهت تنظیم پارامترهای PSO از کنترلر فازی استفاده شده است.

ورودیهای سیستم فازی عبارتند از :

F_{best} : بهترین تابع شایستگی است که توسط گروه در تکرار qام بدست آمده است.

P-dist: نرم فاصله pbest ذره iام تا Gbestکل ذراتمی باشد.

UN : تعداد تکرارهایی است که موقعیت ذره تغییری ننموده است

خروجیهای سیستم فازی عبارتند از:

Wi: پارامتر اینرسی وزن ذره می باشد

C1,i: پارامتر شناختی ذره iام می باشد

C2,i: پارامتر اجتماعی ذره iام می باشد

در مسائلی که هدفشان یافتن مینیمم است مقدار پایین fbest بدین مفهوم است که حداقل یکی از ذرات به ناحیه مناسب از فضای جستجو رسیده است و می تواند ذرات را به سمت ناحیه مناسب راهنمایی نماید. مقدار بالای fbest بدین مفهوم است که ذرات هنوز به ناحیه مناسب را نیافته اند و هر ذره باید به تنهایی به جستجوی محلی بپردازد. UN تعداد تکرارهایی است که هیچ بهبودی در موقعیت ذره ایجاد نشده است و جستجوی محلی ذره بی فایده است و ذره جهت بهبود موقعیت خود باید به جمع نگاه کند. P_{dist} می تواند معیاری از تنوع باشد . مقدار زیاد این پارامتر بدین مفهوم است که تنوع در بین ذرات زیاد است و مقدار کم این پارامتر بدین مفهوم است که تنوع در بین ذرات کم است P_{dis} با استفاده از فرمول(۳۴)می شود

$$P_{dist} = \| P_g - P_i \| = \sum_{k=1}^{H} | W_{kg} - W_{ki} | \qquad (۳۴)$$

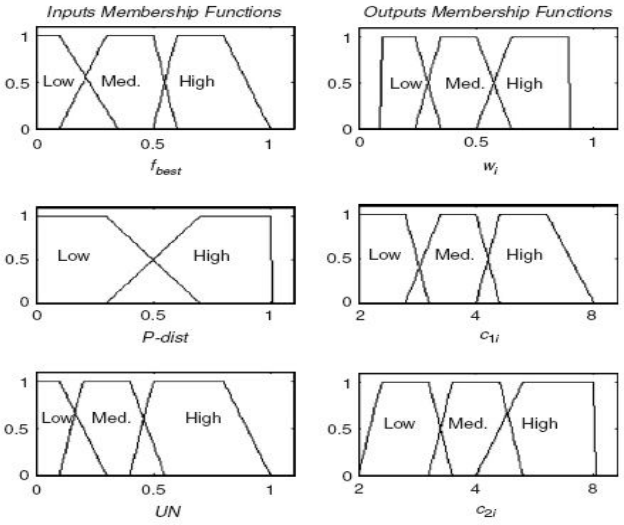
بنابراین می‌توان با استفاده از وضعیتی که ذره در فضای جستجو دارد پارامتر مناسب آن وضعیت را انتخاب نمود. ذره در طول حیات خود در وضعیتهای متفاوتی قرار میگیرد به عنوان مثال هنگامیکه fbest مقدار بالایی دارد و Pdistمقدار کمی است بدین مفهوم است که ذره در مکان نامناسبی قرار دارد و در این حالت باید w را افزایش داد تا سرعت ذره افزایش یافته و ذره محدوده جستجوی خود را افزایش دهد. در چنین وضعیتی می‌توان w را افزایش داد تا سرعت ذره افزایش یافته و ذره بتواند فضای بیشتری را جستجو

نماید. در این مرحله تاکید بر جستجوی عمومی است. می‌توان رفتار اجتماعی ذره را کاهش و رفتار شناختی ذره افزایش می‌یابد.

هنگامیکه UN مقدار بالایی باشد و fbest ذره مقدار متوسطی باشد بدین مفهوم است که ذره مینیمم محلی قرار گرفته است و هنگامیکه fbest مقدار بسیار پایینی باشد و UN مقدار بالایی باشد بدین مفهوم است که ذره به ناحیه نهایی و جواب رسیده است بنابراین باید w و پارامترهای شناختی را کاهش داد و رفتار اجتماعی را افزایش داد تا ذرات سریعا به آن ناحیه همگرا شوند. شش قانون فازی طبق معادله(۳۵)ایجاد شده است:

(1)IF f_{best} is low and P-dist is low THEN o_i is high, c_{1i} is high, and c_{2i} is low. (2) IF f_{best} is medium and UN is high and P-dist is high THEN o_i is medium c_{1i} is high, and c_{2i} is medium. (3) IF f_{best} is medium and UN is low and P-dist is high THEN o_i is medium, c_i is medium, and c_{2i} is medium. (4) IF f_{best} is high and UN is high THEN o_i is low, c_{1i} is low, and c_{2i} is high. (5) IF f_{best} is low and UN is high THEN o_i is high, c_{1i} is low, and c_{2i} is high. (6) IF f_{best} is high and P-dist is high THEN o_i is low, c_{1i} is low, and c_{2i} is high.	۵)

توابع عضویت فازی طبق شکل ۰-۹ ایجاد شده اند.

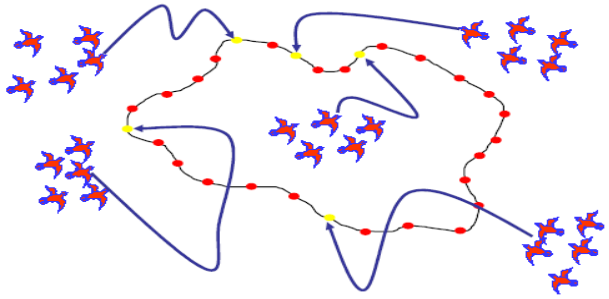


شکل ۰-۹) توابع عضویت فازی در PSO هوشمند

این روش نسبت به PSO استاندارد راندمان بالاتری دارد اما مشکل این روش در این است که در مینیمم ایجاد قوانین باید در این روش به صورت دستی باشد. کما اینکه این قوانین بهترین قوانین فازی نمیباشند.

۱.۳.۱.۳.۶ بهینه سازی گروه ذرات مبتنی بر زیر گروه ها (چند جمعیتی)

می توان بجای استفاده از یک گروه از ذرات می‌توان از مجموعه‌ای از گروه ها استفاده نمود. ایده اصلی این است که یک گروه را به مجموعه‌ای از گروه‌ها تقسیم نموده که هر گروه بهترین ذرات عمومی خود را دارا می‌باشد. تنها ارتباط بین ذرات زمانی رخ می‌دهد که اپراتور تولید نسل در زمان انتخاب ذرات، ذراتی را انتخاب نماید که در دو گروه مجزا قرار دارند.



شکل ۰-۱۰): مثالی از الگوریتم چند جمعیتی برای تشخیص لبه

در [۳۴] نتایج نشان داده است که کارایی این روش بالاتر از PSO نمی‌باشد. در این آزمایش یک گروه با ۲۰ ذره را به شش گروه تقسیم کرده و هر گروه شامل ۴ ذره است .گروه‌های با تعداد کم دارای تنوع کمی بوده و بنابراین قدرت جستجوی عمومی پایینی دارند. هیچ رویدادی از شبیه عمل کردن ذرات یک گروه پیش‌گیری نمی‌نماید. برای رفع این مشکل یک راه حل جدید ارائه شده است :

از دو گروه که هر کدام دارای ۲۰ ذره می‌باشند استفاده شده است این دو گروه در فضای جستجو از یکدیگر دور نگه داشته می‌شوند. ذرات درون این دو گروه توسط اپراتور mutation در فضای جستجو قرار داده می‌شوند. راه کار استفاده از اپراتور mutation کارایی بالاتری نسبت به PSO دارد. عمده ترین مشکل این روش زمانی است که تصمیم می‌گیرند که آیا دو گروه به یکدیگر نزدیک هستند یا خیر که این مشکل بسیار وابسته به مسأله می باشد.

۱.۳.۱.۳.۷ مدل Meta PSO

MPSO ساده‌ترین روش استفاده از چند گروه ذره می‌باشد. هر ذره دارای دو اندیس است، اندیس اول $j=1,\ldots,N_s$ است که گروهی را مشخص می نماید که ذره به آن تعلق دارد و اندیس دوم $i=1,\ldots,N_{pj}$ است که اندیس ذره در گروه را مشخص می نماید. جهت سادگی کار تعداد ذرات گروه‌ها ، با یکدیگر برابر در نظر

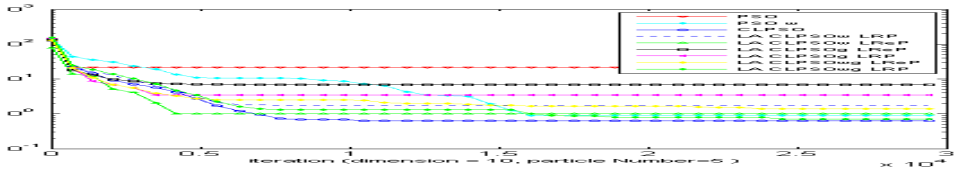
گرفته می شود. $N_p = N_{pj} \ \forall \ j = 1,..., N_s$. بردار سرعت در الگوریتم MPSO به صورت معادله(۳۶) به روز می گردد:

$$V^d_{j,i+1} = w_{j,i} * V^d_{j,i} + c_1 * rand1_{j,i} * (P^d_{j,i} - X^d_{j,i}) + c_2 * rand2_{j,i} * (G^d - X^d_{j,i}) + c_3 * rand3_{j,i} * (S^d_j - X^d_{j,i}) \quad (۳۶)$$

Pj,i بهترین موقعیتی است که ذره i موجود در گروه jام در دوره حیات خود یافته است. Sj بهترین موقعیتی است که توسط گروه j یافته شد است. G بهترین موقعیتی است که بین کلیه گروه ها یافته شده است. سایر مقدار دهی ها کاملا شبیه به الگوریتم PSO استانداردمی باشد.

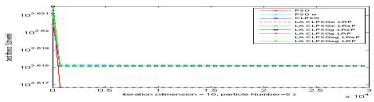
۱.۳.۱.۳.۸ مدل Modified Meta PSO

الگوریتم M^2PSO بهبود یافته الگوریتم MPSO و هدف آن جدا نگه داشتن گروه ذرات از یکدیگر می-باشد. بردار سرعت در این الگوریتم به صورت معادله(۳۷) است.



(۳۷)

این فرمول کاملا شبیه فرمول سرعت در MPSO می باشد با این تفاوت که به انتهای فرمول سرعت آن عبارت دیگری افزوده شده است. این عبارت مجموع دور بودن یک ذره در گروه نسبت به سایر ذرات گروه های دیگر را مشخص می نماید.



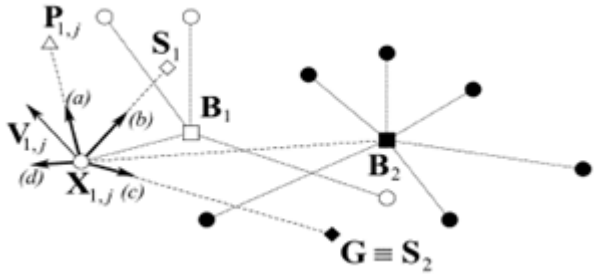
(۳۸)

ξ یک وزن با مقدار ثابت می باشد و φ یک وزن است که به صورت تصادفی مقدار دهی می‌گردند. نیروی دور نگه داشتن یا جدا نگه داشتن ذرات[۴۳] با تابعی از فاصله معرفی می گردد که قدرت آن با γ نمایش داده می‌شود. در بیشتر موارد مقدار γ=۲ در نظر گرفته می شود.

[43] Repulsive

۱.۳.۱.۳.۹ مدل Stabilized Modified Meta PSO

یک روش جهت بهبود M^2PSO این است که گروهی که بهترین ذره آن به عنوان بهترین ذره در بین سایر گروه‌ها نیز محسوب می‌شود Sj=G، از سایر گروه های دیگر دور نگه داشته نشوند، به بیان دیگر این گروه باید موقعیت خود را حفظ کرده و بدون تغییر باقی بماند. این امر باعث می شود که بهترین گروه بتواند جستجوی خود را در اطراف بهترین جواب دنبال نموده و از مینیمم محلی بیرون بیاید. در شکل ۰-۱۱ نمونه ای از این الگوریتم به صورت گرافیکی نمایش داده شده است، که جهت سادگی دو گروه در این شکل نمایش داده شده اند. گروه اول با ذرات سپید و گروه دوم با ذرات سیاه نشان داده شده اند. نیروهای ذره $X_{1,j}$ با سرعت $V_{1,j}$ در شکل کشیده شده است.



شکل ۰-۱۱) مقایسه SM^2PSO با سایر الگوریتم های PSO

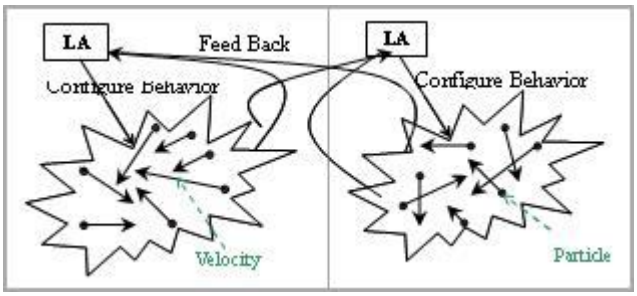
این الگوریتم نسبت به سایر الگوریتم های PSO, CPOS-H2 راندمان بالاتری دارد

۱.۳.۱.۳.۱۰ مدل CLAPSO[۴۴]

CLAPSO تر کیبی از اتوماتای یادگیر سلولی و بهینه سازی گروه ذرات میباشد. در این مدل، هر سلول اتوماتای یادگیر سلولی دارای یک جمعیت از ذرات و همچنین یک اتوماتای یادگیر است. مانند مدل حرکت دسته جمعی ذرات، هر یک از ذرات دارای یک بردار موقعیت و یک بردار سرعت هستند. اتوماتای یادگیر هر سلول دارای دو عمل «دنباله روی» و «ادامه مسیر فعلی» میباشد و وظیفه تنظیم رفتار ذرات همان سلول را بر عهده دارد. در الگوریتم پیشنهادی توسط اتوماتای یادگیر انجام میشود که دو مزیت مهم دارد: اولا میتوان از دانش موجود مساله در تعیین روند تغییرات وزن میانی استفاده نمود و ثانیا این روند با گرفتن بازخورد از اجری الگوریتم اصلاح میگردد. جهت تنظیم رفتار ذرات، اتوماتای یادگیر هر سلول، از تجربیات خود و اتوماتاهای یادگیر سلولهای همسایه، استفاده میکند. بدین ترتیب نوعی همکاری، بین جمعیتهایی یک سلول

[44] Cellular Learnin Automata Particle Swarm Optimization

و سلولهای همسایه شکل میگیرد که باعث عملکرد بهتر آنها میگردد شکل ۰-۱۲ نمای کلی یک CLAPSO با دو سلول را نشان میدهد.



شکل ۰-۱۲) شمایی ییک CLA-PSO با دو سلول

روش کار الگوریتم پیشنهادی به این صورت است که در ابتدا، تعداد سلولها و نوع همسایگی اتوماتای یادگیر سلولی مشخص میشود و موقعیت و سرعت ذرات و همچنین احتمال انتخاب اعمال اتوماتای سلولها مقداردهی اولیه میشوند. سپس تا زمانیکه حداکثر تعداد گامها انجام گردد و یا هدف مورد نظر حاصل شود، مراحل زیر تکرار میشوند:

اتوماتای یادگیر یکی از اعمالشان را با توجه به بردار احتمال اعمال، انتخاب میکنند.

با توجه به عمل انتخاب شده توسط اتوماتای یادگیر هر سلول، نحوه بروزرسانی سرعت ذرات در آن سلول مشخص میشود و ذرات، سرعت و موقعیت خود را به روز میکنند. در صورت انتخاب عمل «دنباله روی»، تنها دنبال کردن بهترین تجربه شخصی و بهترین تجربه گروهی، در به روز نمودن سرعت ذرات مد نظر قرار خواهند گرفت و از سرعت فعلی ذرات صرفنظر میشود. در صورت انتخاب عمل «ادامه مسیر فعلی»، سرعت جدید ذرات برابر با سرعت فعلی آنها خواهد بود و ذره همچنان مسیری که میرفت ادامه خواهد داد.

بر اساس نتایج به روزرسانی موقعیت ذرات و همچنین مقایسه وضعیت بهترین ذره جمعیت سلول با بهترین ذره های سلولهای همسایه، عمل اتوماتای یادگیر ، ارزیابی میشود و بردار احتمال انتخاب اعمال اتوماتای یادگیر اصلاح میشود.

برای ارزیابی عمل انتخاب شده، به این صورت عمل میشود که موقعیت هر ذره در اثر انجام یک گام، با موقعیت قبلیش مقایسه میشود. چنانچه درصد مشخصی از جمعیت موقعیتشان بهبود یافته بود، عمل انتخاب شده مثبت ارزیابی میشود و در غیر اینصورت عمل انتخاب شده منفی ارزیابی خواهد شد. اگر عمل انتخاب شده، «دنباله روی» باشد، برای مثبت ارزیابی شدن عمل، بایستی موقعیت cfimp درصد جمعیت بهبود یافته باشد و چنانچه عمل انتخاب شده، «ادامه مسیر فعلی» باشد، بهبود یافتن موقعیت cgimp درصد جمعیت برای مثبت ارزیابی شدن عمل لازم است.

برای تعیین مقادیر cf_{imp} و cg_{imp}، در ابتدا gbest جمعیت سلول با gbest جمعیتهای سلولهای همسایه مقایسه میشود. هر چه در این مقایسه، gbest جمعیت سلول وضعیت بهتری داشته باشد، استنباط میگردد که بهترین ذره جمعیت به نسبت بهترین ذرات جمعیتهای سلولهای همسایه در قسمت مناسبتری از فضای جستجو قرار گرفته است و مقدار cf_{imp} کمتر و مقدار cg_{imp} بیشتر قرار داده میشود. در واقع سعی میگردد تا عمل «دنباله روی» تشویق گردد. بر عکس، چنانچه در مقایسه انجام شده، gbest وضعیت بدتری داشته باشد، سعی میشود تا با تشویق عمل «ادامه مسیر فعلی»، قسمتهای بهتری از فضای جستجو کشف شوند و به همین دلیل مقدار cf_{imp} بیشتر و مقدار cg_{imp} کمتر قرار داده میشود.

برای استفاده از الگوریتم، مجموعه ای از پارامترها باید تنظیم شوند. ابتدا باید شرایطی مشخص و دقیق تعیین گردند که کیفیت gbest جمعیت سلول در مقایسه با سلولهای همسایه، توسط آنها سنجیده میشود. برای مثال اگر سلولی ۸ همسایه داشته باشد و m تعداد سلولهای همسایه ای باشدکه gbest جمعیتشان بدتر از gbest جمعیت سلول مورد بحث باشد، آنگاه، یک مجموعه شرایط میتوانند به صورت مجموعه معادلات۳۸ باشند.

$$Stat = \begin{Bmatrix} good & m > 6 \\ avg & 3 \le m \le 6 \\ bad & m < 3 \end{Bmatrix} \qquad (۳۹)$$

پس از تعیین این شرایط، برای هر دسته وضعیت، باید مقادیر دقیق cf_{imp} و cg_{imp} مشخص گردند. مثلا میتوان برای دسته خوب، cf_{imp} را برابر با ۳۰ و cg_{imp} را برابر با ۸۰ قرار داد. در صورت تنظیم پارامترها به صورت گفته شده الگوریتم به این صورت عمل خواهد کرد: اگر بهترین ذره جمعیت سلول در مقایسه با بهترین ذرات جمعیتهای سلولهای همسایه، حداقل از ۶ تای آنها بهتر باشد، آنگاه، اگر عمل «دنباله روی» انتخاب شود، فقط کافیست تا موقعیت ۳۰ درصد از ذرات را بهبود دهد تا مثبت ارزیابی شود ولی اگر عمل «ادامه مسیر فعلی» انتخاب شود، برای مثبت ارزیابی شدن، باید موقعیت ۸۰ درصد ذرات را بهبود دهد.

۱-۳-۱-۴- انواع بهینه سازی گروه ذرات همکار Cooperative PSO

در این بخش نوعی از الگوریتم های بهینه سازی گروه ذرات معرفی می شود که ذرات با یکدیگر به گروه هایی تقسیم شده و گروه ها در رسیدن به جواب بایکدیگر در تعاملند.

۱.۳.۱.۴.۱ بهینه سازی گروه ذرات با چند فاز Multi Phase

در Multi phase ذرات را به گروههایی تقسیم می نماید که هر گروه وظیفه انجام یک کار را به عهده داشته و یک رفتار به خصوص را از خود بروز می دهد. رفتار و اعمال یک ذره در طی تکرار الگوریتم تغییر می نماید.و این امر باعث می گردد که ذرات از یک گروه به گروه دیگر مهاجرت نمایند. مدل جذبی و دفعی ARPSO و مدل MP-PSO[۵][45] ، DLPSO[46] [۳۶]، LCPSO[47] نمونه هایی از این الگوریتم ها می باشند. که در ادامه به چند نمونه ار آنها می پردازیم:

۱.۳.۱.۴.۲ مدل جذبی و دفعی۴۸(ARPSO):

این مدل جذبی دفعی در واقع تناوبی است بین دو فاز جذب و دفع. در فاز جذب ARPSO از PSO استفاده می‌کند تا به جریان اطلاعات سرعت ببخشد ، ذرات یکدیگر را جذب نمایند و این امر باعث کاهش تنوع گردد. در فاز دفع ذرات از بهترین راه حل پیدا شده دور می‌گردند و این امر باعث افزایش تنوع می‌گردد. در واقع در این فاز جهت سرعت ذرات معکوس می شود و هر ذره توسط بهترین ذره گروه $\vec{g}(t)$و بهترین موقعیت خود ذره $\vec{p}(t)$ دفع می گردد. معادله به روز شدن سرعت ذرات به صورت زیر می باشد:

$$\vec{v}(t+1) = w\vec{v}(t) - \phi_1\left(\vec{p}(t) - \vec{x}(t)\right) - \phi_2\left(\vec{g}(t) - \vec{x}(t)\right) \qquad (۴۰)$$

هر گاه ذرات در فاز جذب باشند رفته رفته تنوع کاهش می یابد به گونه ای که تنوع به کمترین مقدار خود یعنی d_{low} می رسد. در این زمان است که از فاز دفع استفاده می شود و ذرات از یکدیگر دور می شوند. در این حالت تنوع افزایش می یابد به حدی که به d_{heigh} می رسد و دوباره باید از فاز جذب استفاده نمود. الگوریتم این روش به صورت

```
Program PSO
    init();
   while not done do
setDirection(); // new!
updateVelocity();
newPosition();
assignFitness();
Function setDirection
```

[45] Multi-phase PSO
[46] Devision of Labor PSO
[47] Life Cycle PSO
[48] Attractive and Repulsive PSO(ARPSO)

```
if (dir > 0 && diversity < dLow) dir = -1;
if (dir < 0 && diversity > dHigh) dir = 1;
```

شکل ۰-۱۳نمایش داده شده است:

```
Program PSO
    init();
   while not done do
setDirection(); // new!
updateVelocity();
newPosition();
assignFitness();
Function setDirection
if (dir > 0 && diversity < dLow) dir = -1;
if (dir < 0 && diversity > dHigh) dir = 1;
```

شکل ۰-۱۳) شمای الگوریتم ARPSO

setDirection مشخص می‌نمایدکه کدام فاز فعال است. تابع calculate Dirversity تنوع ذرات را اندازه گیری می نماید. این تنوع به صورت زیر محاسبه می شود:

$$diversity\ (s) = \frac{1}{|S|.|L|}.\sum_{i=1}^{|s|}\sqrt{\sum_{j=1}^{N}(p_{ij}-p_j)^2} \qquad (۴۱)$$

S مشخص کننده گروه ، |S| تعداد ذرات گروه ، |L| طول بلند ترین همسایگی در فضای جستجو را نشان میدهد و N تعداد ابعاد فضای جستجو می باشد. p_{ij} مشخص کننده بعد jام از ذره iام در گروه می باشد. $\overline{p_j}$ بهترین موقعیت ذره در بعد Jام می باشد. توجه شودکه تنوع بسیار به اندازه ذرات گروه بستگی دارد.می توا ن معادله سرعت در دو فاز جذب و دفع را با یکدیگر به صورت زیر ادغام نمود:

$$\vec{v}(t+1) = w\vec{v}(t) - div\,(\phi_1\left(\vec{p}(t)-\vec{x}(t)\right)-\phi_2\left(\vec{g}(t)-\vec{x}(t)\right)\ ) \qquad (۴۲)$$

نتایج نشان می دهندکه راندمان این روش نسبت به PSO و GA راندمان بهتری است.

۱.۳.۱.۴.۲.۱ مدل چرخه زندگی[49]:

الگوریتم جستجوی اکتشافی خود تطبیقی [50] را مدل چرخه زندگی نام می‌نهند. مدل چرخه زندگی ترکیبی از PSO و الگوریتم ژنتیک با روش تپه نوردی می باشد. انگیزه این روش ادغام مزایای PSO و الگوریتم ژنتیک با دید تپه نوردی در یک الگوریتم واحد می‌باشد. در این الگوریتم هر عضو (نمایش راه

[49] Lifecycle
[50] Self-adaptive heuristic search

حلهای پتانسیلی) به عنوان ذرات PSO شروع به کار می‌نماید و سپس با توجه به کاراییشان (در جستجوی راه حل) می‌توانند به عنوان اعضای الگوریتم ژنتیک یا تپه نوردی در نظر گرفته شوند و سپس دوباره به عنوان ذرات PSO در نظر گرفته شوند. این کار آنقدر تکرار می‌گردد تا همگرایی اتفاق بیافتد. مدل چرخه زندگی با روشهای GA،PSO، تپه نوردی مقایسه شده و نشان داده شده است که برای پنج آزمایش خاص کارایی PSO در سه آزمایش بهتر از روش چرخه زندگی بوده است.

۱.۳.۱.۴.۲.۲ ساختار خوشه‌ای Cluster Formation

در این گونه از الگوریتم ها ذرات را به خوشه‌هایی تقسیم نموده که ذرات موجود در هر خوشه به جستجوی محلی در آن خوشه می پردازند . هر ذره به سمت میانگین بهترین موقعیت ملاقات شده توسط ذرات در خوشه ای که در آن قرار دارد حرکت می نماید.

۱.۳.۱.۴.۲.۳ بهینه سازی گروه ذرات همکاری چند بخشی Cooperative Split PSO(CPSO)

به جای استفاده از یک گروه که دارای s ذره است و هدفشان یافتن جواب بهینه بردار n بُعدی است، به ازای هر بُعد یک گروه ذرات انتخاب می‌شود، بنابراین n گروه ذره استفاده می شود. که هر گروه خود دارای s ذره می باشد. باید توجه شود که تابعی که باید بهینه شود به یک بردار n بُعدی نیاز دارد، که مشکلات زیر است: انتخاب: بردار راه حلها به n قسمت تقسیم شده و به هر قسمت یک گروه با m ذره تخصیص داده می-شود. برای همکاری بین گروه ها به s*n ترکیب ذرات نیاز می باشد. ساده ترین راه حل این است که بهترین ذره در هر گروه انتخاب شده و با یکدیگر همکاری نمایند. البته این روشی حریصانه است.

حل این مشکل پاسخی است به سوال زیر"هر ذره به چه میزان در جواب تاثیر دارد؟" هر گروه ترکیب گروه ها به یک جواب بهینه می رسد، هر گروه به چه میزان از این پاسخ تاثیر می‌گیرد؟ یکی از ساده ترین راه حلها این است که هر گروه باید دارای یک مقدار اعتباری باشد که به اندازه آن اعتبار، تاثیر بگیرد.

تفاوت عمده CPSO, CCGA در این است که در CPSO که تعامل اجتماعی ذرات در گروه عملیات بهینه سازی را میسر می‌سازد در حالیکه در CCGA جابجایی اطلاعات ژنتیکی باعث این عمل می گردد. فرض شود که ذره i در گروه، دارای سرعت، موقعیت و بهترین موقعیت ملاقات شده باشد. هر ذره به تنهایی یک بردار n بُعدی را نمایش می دهد که معرف یک پاسخ یا راه حل برای مساله می باشد. گاهی امکان دارد که قسمت هایی از این بردار به پاسخ صحیح نزدیک شده باشند در حالیکه قسمتهای دیگر بردار از پاسخ صحیح دور باشند. بنابراین در کل این ذره مناسب به نظر نمی‌رسد و باید به موقعیت بهتری برود . امکان دارد

که آن قسمتهایی از بردار ذره که به جواب نزدیک بوده‌اند در موقعیت جدید، از پاسخ صحیحی که در در مرحله ثبل بدست آورده اند فاصله بگیرند. برای رفع این مشکل یکی از این روشها این است که فرکانس اندازه گیری تابع ملاک شایستگی افزایش داده می شود. به عنوان مثال هر لحظه که یکی از ابعاد بردار ویژگی به روز میشود. تابع ملاک شایستگی نیز اندازه‌گیری شود. اما با این روش هنور مشکل باقی است و به طور کامل حل نشده است. ارزیابی ملاک شایستگی فقط با داشتن یک بردار n بعدی میسر است. بنابراین با به روز شدن یکی از ابعاد بردار ویژگی n-1 بعد دیگر از بردار ویژگی را باید به روز نمود. در ادامه به بررسی الگوریتم CPSO می‌پردازیم.

۱.۳.۱.۴.۲.۳.۱ *الگوریتم CPSO_Sk:*

در PSO استاندارد بردار ویژگی n بُعدی می‌باشد. می‌توان این بردار n بعدی را به n گروه ذره یک بعدی تقسیم نمود. هر گروه از ذرات وظیفه به روز نمودن یک بعد از بردار ویژگی‌ها را به عهده دارد. این تقسیم-بندی شبیه تقسیم‌بندی است که در روش relaxation method بکار رفته است. یکی از مشکلات این روش این است که تابعی که باید بهینه عمومی آن یافته شود تابعی n بُعدی است. اگر هر گروه به تنهایی فضای جستجوی یک بعد را جستجو نمایند مشخص است که محاسبه تابع ملاک شایستگی امکان پذیر نمی‌باشد. لذا نیاز به یک بردار مفهموی[۵۱] است تا بتوان به راحتی ذرات را ارزیابی نمود. ساده ترین راه حل برای این هدف این است که از هر گروه بهترین ذره را به عنوان نماینده آن بُعد انتخاب نموده و با ترکیب این ذرات بردار ویژگی n بُعدی تولید شود. به منظور ارزیابی گروه j ام ، n-1 بردار دیگر ثابت در نظر گرفته شده و به ازای هر کدام از ذرات گروه‌jام ، فقط همان بُعد jام تغییر داده می‌شود. اولین الگوریتم CPSO_S که توسط Van den Bergh , Engelbrecht معرفی شده است درشکل ۰-۱۴ آورده شده است. این الگوریتم فضای جستجو را به n زیر فضای جستجو تقیسم می کند. در این الگوریتم $P_j.x_i$ موقعیت ذره iام در گروه jام را مشخص می‌نماید که در واقع jامین بعد از بردار ویژگی nبعدی را مشخص می‌نماید. $P_j.\hat{y}.$ موقعیت بهترین ذره در گروه nام را مشخص می‌نماید. b(j,z) بردار ویژگیn بعدی است که از ترکیب موقعیت بهترین ذره n گروه تشکیل شده است. بجز بعد jام از این بردار که مقدار ذره zام گروه Pj در آن قرار دارد. یکی از مزایای این الگوریتم این است که به ازای هر یک از ابعاد بردار که به روز می‌شود، تابع خطای f نیز ارزیابی می‌شود. بهترین بردار context جاری توسط $b(l,P_l.\hat{y})$ مشخص می‌گردد. هر ذره به عنوان مثال یک فضای جستجوی

[51] Context Vector

۳۰ بعدی در الگوریتم CPSO-S دارای ۳۰ گروه ذره یک بعدی می‌باشد. در طول یک تکرار از الگوریتم 30*30=900 ترکیب مختلف می توان ایجاد کرد ،

از مزایای این الگوریتم این است که در هر مرحله فقط یک بعد از بردار ویژگی به روز می شود و همچنین در این الگوریتم تنوع راه حل ذرات تفاوت قابل ملاحظه ای به PSO دارد.

define
$\mathbf{b}(j,z) \equiv (P_1.\hat{\mathbf{y}}, P_2.\hat{\mathbf{y}}, \ldots, P_{j-1}.\hat{\mathbf{y}}, z, P_{j+1}.\hat{\mathbf{y}}, \ldots, P_n.\hat{\mathbf{y}})$
Create and initialise n one-dimensional PSOs : $P_j,\ j \in [1..n]$
repeat:
  for each swarm $j \in [1..n]$:
    for each particle $i \in [1..s]$:
      if $f(\mathbf{b}(j,P_j.\mathbf{x}_i)) < f(\mathbf{b}(j,P_j.\mathbf{y}_i))$
        then $P_j.\mathbf{y}_i = P_j.\mathbf{x}_i$
      if $f(\mathbf{b}(j,P_j.\mathbf{y}_i)) < f(\mathbf{b}(j,P_j.\hat{\mathbf{y}}))$
        then $P_j.\hat{\mathbf{y}} = P_j.\mathbf{y}_i$
    endfor
    Perform PSO updates on P_j
  endfor
until stopping condition is true

شکل ۰-۱۴): الگوریتم CPSO-S

۱.۳.۱.۴.۲.۳.۲ *الگوریتم CPSO-Sk:*

گاهی اوقات که بعضی از ابعاد به یکدیگر وابسته می باشند آن ابعاد را می توان به عنوان یک گروه در نظر گرفت. بعضی از گروه ذرات یک بعد و بعضی دیگر c بعدی می باشند.($c < n$). این الگوریتم که $CPSO\text{-}S_k$ نامیده می شود از k گروه ذره استفاده می نماید که ($k<n$) است. در واقع CPSO-S نوعی از CPSO-Sk می باشد که در آن k=n است.به عدد k درCPSO-Sk فاکتور تقسیم[۵۲] گفته می شود. شکل ۰-۱۵ الگوریتم CPSO-Sk را نمایش می‌دهد. توجه شود که در CPSO-Sk می توان گونه های مختلفی از PSO را به کار برد.

[52] Split Factor

define
$\mathbf{b}(j,\mathbf{z}) \equiv (P_1.\hat{\mathbf{y}},\ldots,P_{j-1}.\hat{\mathbf{y}},\mathbf{z},P_{j+1}.\hat{\mathbf{y}},\ldots,P_K.\hat{\mathbf{y}})$
$K_1 = n \bmod K$
$K_2 = K - (n \bmod K)$
Initialise K_1 $\lceil n/K \rceil$-dimensional PSOs:
$P_j,\ j \in [1..K_1]$
Initialise K_2 $\lfloor n/K \rfloor$-dimensional PSOs:
$P_j,\ j \in [(K_1+1)..K]$
repeat:
  for each swarm $j \in [1..K]$ **:**
    for each particle $i \in [1..s]$ **:**
      if $f(\mathbf{b}(j,P_j.\mathbf{x}_i)) < f(\mathbf{b}(j,P_j.\mathbf{y}_i))$
        then $P_j.\mathbf{y}_i = P_j.\mathbf{x}_i$
      if $f(\mathbf{b}(j,P_j.\mathbf{y}_i)) < f(\mathbf{b}(j,P_j.\hat{\mathbf{y}}))$
        then $P_j.\hat{\mathbf{y}} = P_j.\mathbf{y}_i$
    endfor
    Perform PSO updates on P_j
  endfor
until stopping condition is true

شکل ۰-۱۵): الگوریتم CPSO-S_k

۱.۳.۱.۴.۲.۳.۳ *الگوریتم CPSO-Hk:*

الگوریتم CPSO-S_K امکان دارد که در مینیمم محلی قرار بگیرد هر چند این روش نسبت به روشPSO استاندارد سرعت همگرایی بالاتری دارد. الگوریتم استاندارد PSO عیب این روش را ندارد. برای رفع این مشکل، الگوریتم جدید CPSO-Hk که ترکیبی از الگوریتم CPSO-Sk و الگوریتم استاندارد PSO ایجاد شد. این دو الگوریتم را به دو روش می توان با یکدیگر ترکیب نمود. یکی از راه های ترکیب این دو الگوریتم این است که الگوریتم CPSO-Sk عمل نماید و هنگامیکه در pseudominimizer قرار گرفت از الگوریتم استاندارد PSOاستفاده کند. سوییچ کردن بین این دو الگوریتم کار چندان ساده ای نمی باشد.

define
$\mathbf{b}(j,\mathbf{z}) \equiv (P_1.\hat{\mathbf{y}},\ldots,P_{j-1}.\hat{\mathbf{y}},\mathbf{z},P_{j+1}.\hat{\mathbf{y}},\ldots,P_K.\hat{\mathbf{y}})$
$K_1 = n \bmod K$
$K_2 = K - (n \bmod K)$
Initialise K_1 $\lceil n/K \rceil$-dimensional PSOs:
$P_j,\ j \in [1..K_1]$
Initialise K_2 $\lfloor n/K \rfloor$-dimensional PSOs:
$P_j,\ j \in [(K_1+1)..K]$
Initialise an n-dimensional PSO : Q
repeat:
  for each swarm $j \in [1..K]$ **:**
    for each particle $i \in [1..s]$ **:**
      if $f(\mathbf{b}(j,P_j.\mathbf{x}_i)) < f(\mathbf{b}(j,P_j.\mathbf{y}_i))$
        then $P_j.\mathbf{y}_i = P_j.\mathbf{x}_i$
      if $f(\mathbf{b}(j,P_j.\mathbf{y}_i)) < f(\mathbf{b}(j,P_j.\hat{\mathbf{y}}))$
        then $P_j.\hat{\mathbf{y}} = P_j.\mathbf{y}_i$
    endfor
    Perform PSO updates on P_j
  endfor
  Select random $k \sim U(1,s/2) \mid Q.\mathbf{x}_k \neq Q.\hat{\mathbf{y}}$
  $Q.\mathbf{x}_k = \mathbf{b}(1,P_1.\hat{\mathbf{y}})$
  for each particle $j \in [1..s]$ **:**
    if $f(Q.\mathbf{x}_j) < f(Q.\mathbf{y}_j)$
      then $Q.\mathbf{y}_j = Q.\mathbf{x}_j$
    if $f(Q.\mathbf{y}_j) < f(Q.\hat{\mathbf{y}})$
      then $Q.\hat{\mathbf{y}} = Q.\mathbf{y}_j$
  endfor
  Perform PSO updates on Q
  for swarm $j \in [1..K]$ **:**
    Select random $k \sim U(1,s/2) \mid P_j.\mathbf{y}_k \neq P_j.\hat{\mathbf{y}}$
    $P_j.\mathbf{x}_k = Q.\hat{\mathbf{y}}_j$
  endfor
until stopping condition is true

شکل ۰-۱۶): الگوریتم $\mathbf{CPSO\text{-}H_k}$

هر دو الگوریتم در هر تکرار به طور متناوب اجرا شوند و بهترین راه حلهایی را که کشف می نمایند با یکدیگر ردو بدل نمایند. این اطلاعات در واقع همکاری بین این دو الگوریتم را باعث می‌شوند. توجه شود که این همکاری در واقع مانند استفاده مشترک از یک تخته سیاه می باشد که Clearwater در [۳۷] توضیح داده است. یکی از راه حلهای دیگر برای پیاده‌سازی رد و مدل اطلاعات بین دو الگوریتم این است که بعضی از ذرات یکی از الگوریتم‌ها را با تعدادی از بهترین ذرات الگوریتم دیگر جابجا نمود. الگوریتم CPSO-Hk را درشکل ۰-۱۶ این روش نمایش می‌دهد.

بعد از هر تکرار $b(1, P_1.\hat{y})$ بردار context مربوط به الگوریتم CPSO-Sk ، جایگزین بعضی از ذرات الگوریتم استاندارد PSO می شوند. ذراتی از الگوریتم PSO که جایگزین می شوند به طور تصادفی انتخاب می شوند. توجه شود که در این الگوریتم بهترین ذرات هر گروه را نباید جایگزین نمود. تجربه نشان می دهد که ردو بدل زیاد اطلاعات باعث کند شدن الگوریتم می گردد. انتخاب ذراتی که باید جایگزین شوند با استفاده از یک توریع یکنواخت باعث می شود که بیشتر ذرات را بتوان جایگزین نمود .

۱.۳.۱.۴.۳ انواع بهینه سازی گروه ذرات رقابتی Competetive PSO

دراین گونه از الگوریتم‌ها ذرات برای زنده ماندن با یکدیگر به رقابت می‌پردازند. رابطه بین ذرات مانند شکار و شکارچی در طبیعت می باشد. در طبیعت هنگامیکه گروهی از جانوران توسط گروه دیگری از جانوران مورد حمله قرار می‌گیرند، گروه مورد حمله قرار داده شده پراکنده می‌شوند و سعی دارند مناطق بهتری بروند که امن بوده و شکارچیی در آن وجود نداشته باشد. این رفتار باعث می گردد که تعادلی بین جستجوی عمومی بر قرار گردد.

Memetic PSO -۵-۱-۳-۱

یکی از معایب الگوریتم‌های PSO این است که این الگوریتم‌ها در مراحل پایانی عملیات بهینه سازی خاصیت جستجوی عمومی خود را از دست می‌دهند و در مینیمم محلی قرار می‌گیرند. بیشتر الگوریتم‌های بهینه سازی خیلی سریع در ناحیه مناسب و امید بخش قرار می‌گیرند، اما توانایی یافتن پاسخ بهینه را ندارند .جهت رفع این مشکل PSO استاندارد از جستجوهای محلی برای تنظیم وزن اینرسی، ضرایب مولفه شناختی و مولفه عمومی و محدود نموده سرعت ذره استفاده شده است و یا بوسیله جستجو های محلی تصادفی[۵۳] مانند CLPSO. GCPSO این مشکل را حل می نمایند. در بیشتر این الگوریتم ها در ابتدا اجازه داده می شود که ذرات نواحی مناسب را کشف نمایند و سپس ذرات به صورت محلی شروع به جستجو نمایند. به عنوان مثال در [۳۸] از الگوریتم تپه نوردی جهت جستجوی محلی استفاده شده است. در ادامه به چند نمونه از الگوریتم ها می پردازیم.

[53] Stochastic Local Search

۱.۳.۱.۵.۱ بهینه سازی گروه ذرات با یادگیری جامع[۵۴] CLPSO

هر چند انواع متفاوتی از PSO ارائه شده است اما همگرایی زودرس برای حل مسائل چند قله‌ای یکی از مشکلات این الگوریتم ها محسوب می شود. در کل در PSO هر ذره برای به روز نمودن سرعت خود از gbest , pbest استفاده می‌نماید. هر چه درصد بیشتری از gbest در نظر گرفته شود، سرعت همگرایی افزایش می یابد. چون تمام ذرات از gbest برای به هنگام نمودن سرعت استفاده می نمایند بنابراین اگر gbest در مینیمم محلی قرار گیرد همه ذرات در مینیمم محلی قرار می گیرند. تابع $F(x) = f([x^1,x^2,x^3,..,x^D])$ مقادیر شایستگی ذرات در هر بعد می‌باشد. یک ذره امکان دارد که در ابعاد خاصی دارای مقدار شایستگی پایینتری باشد. در[۳۹] به منظور استفاده بهتری از اطلاعات ذرات، استراتژی یادگیری جدیدی ارائه شده است بیان شده است که از تمام pbest های ذرات برای به روز نمودن سرعت استفاده می گردد. با این استراتژی تنوع ذرات در گروه حفظ میشود و از همگرایی زودرس جلوگیری میشود. از بین سه نوع مختلف PSO الگوریتمCLPSO دارای کارایی بالاترین می باشند. با توجه به آزمایشاتی که در این مقاله انجام شده است. دراین روش سرعت هر ذره توسط فرمول (۴۳)محاسبه می گردد.

$$V_i^d = w*V_i^d + c*rand_i^d*(pbest_{fi(d)}^d - X_i^d) \quad (۴۳)$$

به ازای هر بعد از یک ذره خاص مقدار $pbest_{fi(d)}$ انتخاب می شود .$f_i = [f_i(1), f_i(2),\ldots,f_i(D)]$ مشخص می کند که در بعد iام از pbest چه ذره‌ای استفاده می گردد. $pbest_{f_i(d)}^d$ معادل pbest یک ذره یا pbest خود ذره به ازای یک بعد خاص می‌باشد. اینکه به ازای هر بعد از pbest چه ذره‌ای استفاده شود با توجه به یک احتمال Pc این عمل صورت می‌گیرد. به ازای ابعاد مختلف می‌توان احتمالهای زیادی استفاده نمود. به ازای هر بعد یک عدد تصادفی استفاده می‌شود اگر این عدد تصادفی از مقدار احتمال Pc_i بزرگتر بود به ازای آن بعد خاص از pbest همان ذره استفاده می‌شود، در غیر اینصورت از pbest سایر ذرات استفاده می‌شود. الگوریتم انتخابی در روش، انتخاب تورنمنت است

در ابتدا دو ذره به طور تصادفی از بین ذرات انتخاب می‌شوند.

مقدار fitness این دو ذره با یکدیگر مقایسه می‌شوند و ذره‌ای انتخاب می‌شود که شایستگی بالاتری دارد. در CLPSO ذره‌ای که شایستگی بیشتری دارد بهتر است بنابراین برای مسائل مینیمم نمودن از مقادیر منفی شایستگی استفاده می‌شود.

---

[54] Comprehensive learning Particle Swarm Optimization

pbest ذره‌ای را که شایستگی بالاتری دارد انتخاب نموده و اگرpbest های انتخابی مانند pbest خود ذره باشد به صورت تصادفی یک ذره از جمعیت را انتخاب نموده و pbest آنرا در همان بعد در نظر می‌گیریم. در شکل ۲ فلوچارت این روش نمایش داده شده است.

تمام $pbest_{f_i}$ ها می‌توانند یک موقعیت جدید را در فضای جستجو پیدا کنند. جخت اطمینان از اینکه ذرات از ذرات خوبی پیروی می نمایند و عمل یادگیری را انجام می دهند و در جایی ثابتی قرار نمی‌گیرند که زمانشان را تلف نمایند، از یک متغیر به نام gap m استفاده شده است که اگر gbest ذرات بیش از mبار تغییر نیافت، fi ذرات دوباره انتخاب می شوند.

در CLPSO یک احتمال یادگیری به نام PC وجود دارد که مقادیر کوچک Pc برای مسائل multimodalمناسب است. اما برای مسائلی unimodal مقادیر مختلف Pc همه نتایج یکسانی خواهد داشت. اما هدف تاثیرCLPSO بر روی مسائل چند قله ای است به ازای ذرات مقادیر مختلف Pc درنظر گرفته می‌شود که هر ذره در دسته های مختلف جستجوی عمومی یا جستجوی محلی قرار می‌گیرد. فرمول انتخاب احتمال Pc برای ذرات گروه به صورت (۴۴)می باشد.

$$Pc_i = 0.05 + 0.45 * \frac{\left(\exp(\frac{10(i-1)}{ps-1}) - 1\right)}{\exp(10) - 1} \qquad (۴۴)$$

۱.۳.۱.۵.۱.۱ بهینه سازی گروه ذرات مبتنی بر گرادیانت

در بیشتر الگوریتم‌های جستجو از اطلاعات گرادیانت برای یافتن مناطق بهینه استفاده می شود. استفاده از گرادیانت قدرت جستجوی محلی را افزایش داده و در عوض قدرت جستجوی عمومی کاهش می-یابد. در نتیجه این الگوریتم-ها به مینیمم محلی بسیار حساس بوده و فقط برای فضاهای جستجوی پیوسته مناسب می‌باشند. در[۴۰] الگوریتمی به نام two-step PSO ارائه شده است در آن هر ذره دو موقعیت جدید را برای خود تخمین می‌زند، یکی موقعیت جدید ذره بر اساس PSO استاندارد مانند معادله(۴۵) دیگری موقعیت جدید ذره بر اساس کوتاهترین اندازه قدم ذره طبق معادله(۴۶) است.

$$x'_i(t+1) = x_i(t) + v_i(t+1) \qquad (۴۵)$$

$$x''_i(t+1) = x_i(t) + \beta v_i(t+1) \quad where \beta \in (0,1) \qquad (۴۶)$$

در هر بعد موقعیت هر ذره بر اساس معادله (۴۷) محاسبه می‌گردد

$$x_i^d(t+1) = \arg\max\left(-\frac{f(x''^d_i(t+1)) - f(x_i^d(t+1))}{\beta}, \frac{f(x'^d_i(t+1)) - f(x_i^d(t+1))}{\beta} \right) \quad (۴۷)$$

این الگوریتم راندمان بالاتری نسبت به PSO استاندارد دارد. Leap Forg PSO نیز گونه دیگری از این دسته الگوریتم‌ها می‌باشد.

۱.۳.۱.۵.۲ الگوریتمهای بهینه سازی گروه ذرات با چندین نقطه شروع[۵۵]

MPSO، مدل توسعه یافته روش GCPSO می باشد.این الگوریتم به این صورت عمل می‌نماید که در مرحله ۱ ذرات گروه به صورت تصادفی مقدار دهی می‌شوند. در مرحله دوم الگوریتم GCPSO را اعمال نموده تا الگوریتم در یک کمینه محلی همگرا شود. موقعیت کمینه محلی را ذخیره می نماید. مراحل ۱ و۲ آنقدر تکرار می شوند تا شرایط توقف برسد. در مرحله ۲ می توان بجای استفاده ازGCPSO از PSO استفاده نمود.

مدلهای مختلفی از MPSO وجود دارد که بیشتر بررسی هایشان بر روی همگرایی GCPSO می‌باشد.

جهت پی بردن به اینکه ذرات همگرا شده اند یا نه ، از تابع موجود در معادله (۴۸) استفاده می شود.

$$fratio = \frac{f(\hat{y}(t)) - f(\hat{y}(t-1))}{f(\hat{y}(t))} \quad (۴۸)$$

اگر f_{ratio} از یک آستانه‌ای که کاربر مشخص نموده کمتر باشد مقدار یک شمارنده افزایش می‌یابد. اگر تعداد شمارنده به یک حد خاصی (آستانه) رسید، گروه همگرا می‌شود. در کل MPSO کارایی بهتری نسبت به GCPSO دارد. کارایی MPSO به تعداد ابعاد تابع بستگی دارد.

۱-۳-۱-۶- چند نقطه بهینه Multi Optimum Static Programming

توزیع اولیه جمعیت در فضای جستجو نقش به سزای در راندمان PSO بازی می‌کند. در این روش سعی شده است که از مقادیر بهینه که در طی الگوریتم توسط کل گروه بدست آمده استفاده نماید. بنابراین ذرات علاوه بر بهترین موقعیتی که توسط کل گروه مشاهده شده استفاده از M موقعیت بهینه ملاقات شده توسط کل گروه استفاده می‌نمایند سرعت ذره در این الگوریتم به صورت(۴۹) بدست می آید:

[55] Multi Start PSO

$$v_{id} = wv_{id} + \sum^{M} c_k \, rand_k \, (P_{ik,d} - x_{id}) \qquad (۴۹)$$

K Pk,idامین موقعیت ملاقات شده بهتر در کل گروه می‌باشد. ck فاکتور انقباض است و M تعداد ذراتی است که بهترین موقعیت را در کل گروه دارند. این روش نسبت به الگوریتم استاندارد PSO راندمان بالاتری دارد.

۱-۳-۱-۷- روشهای دفعی و افزایش تنوع Repelling Methods

در مدل استانداردPSO امکان داردکه ذرات در یک نقطه می‌نیمم محلی همگرا گردندکه این امر باعث کاهش تنوع در بین ذرات می گردد . جهت افزایش تنوع در بین ذرات روشهای چندی ارائه شده است که در ادامه به آنها می پردازیم:

۱.۳.۱.۷.۱.۱ پراکندگی[۵۶] و PSO

به منظور اجتناب از همگرایی زودرس ، در این روش یک جهش تصادفی به PSO اضافه شد. این روش یک آنتروپی منفی را در بین ذرات تصادفی بیان می نماید.

$$\begin{aligned} &if\ (r_1(t) < c_v)\ then\ \ v_{i,j}(t+1) = r_2(t) V_{\max} \\ &if\ (r_3(t) < c_t)\ then\ \ x_{i,j}(t+1) = R(t) \end{aligned} \qquad (۵۰)$$

که , $r_1(t) \sim U(0,1)$, $r_2(t) \sim U(0,1)$ و $r_3(t) \sim U(0,1)$ می باشند و . c_v و c_t فاکتور بی نظمی است که در بازه [1 0] قرار دارند

۱.۳.۱.۷.۲ بهینه سازی گروه ذرات و عامل دیوانگی[۵۷]:

به منظور اجتناب از همگرایی زودرس در سال ۱۹۹۵ یک اپراتور جدید ارائه شد که Craziness نام داشت. هر چند که بعد به این نتیجه رسیدند که این اپراتور به درد نمی‌خورد[۶]. در ادامه این کار در سال ۲۰۰۳ دو مرتبه این اپراتور جدید معرفی گردید. در هر تکرار تعدادی از ذرات گروه که از مرکز گروه دور می‌باشند انتخاب می‌گردند. موقعیت این ذرات به طور تصادفی تغییر می‌یابد و همزمان سرعت‌هایشان نیز توسط مولفه سرعت تغییر می یابد:

---

[56] Dissipative
[57] Craziness

$$v_{i,j}(t+1) = c_1 r_{1,j}(t)(y_{i,j}(t) - x_{i,j}(t)) \quad (۵۱)$$

با توجه به نتایج این اپراتور تأثیر چندانی برروی کارایی PSO ندارد.

۱.۳.۱.۷.۳ بهینه سازی گروه ذرات جدیدNPSO۵۸

تا به حال در الگوریتم‌هایی که از PSO مشاهده نمودیم موقعیت و سرعت هر ذره بر اساس بهترین موقعیت خود ذره و بهترین موقعیت کل ذرات در مرحله قبل بهنگام می‌شد. اما در این الگوریتم ،NPSO ، شرایط بهنگام سازی کمی تغییر کرده است. موقعیت و سرعت هر ذره بر اساس بدترین موقعیت خود ذره و بدترین موقعیت کل ذرات در مرحله قبل بهنگام می‌شد. هدف اصلی این الگوریتم این است که از کلیه موقعیت های بد در فضای جستجو اجتناب گردد. در[۴۱] ادعا شده است که PSO بهتر از NPSO جوابهای بهینه را پیدا می‌کند.

۱.۳.۱.۷.۴ خود سازمانده بحرانی۵۹(SOC PSO)

به منظور افزایش تنوع جمعیت که باعث اجتناب از همگرایی زودرس می‌گردد. یک روش جدید توسعه یافته PSO در سال ۲۰۰۲ ارائه گردید. که SOC PSO نام گرفت. در این الگوریتم یک اندازه‌گیری که بحرانی نامیده می‌شود صورت میگیرد. در این اندازه‌گیری محاسبه می‌گردد که دو ذره تا چه اندازه به یکدیگر نزدیک می باشند. بنابراین برای افزایش تنوع گروه جای این ذرات را عوض می‌نماید. هر کدام از ذرات گروه دارای یک متغیر CL می باشند. یک ذره با درجه بحرانی بالا بحرانش را به همسایه‌های اطرافش نیز انتقال می‌دهد. بدینصورت که متغیرCL همسایه‌هایش را با ۱(یا هر عددی که برنامه نویس مشخص کرده) مقدار دهی می-نماید. سپس ذره بحرانی بودن خود را با تغییر مقدارCL کاهش می دهد. سپس ذره مکان خود را تغییر می دهد. دو نوع مدل مختلف برای تغییر دادن مکان یک ذره وجود دارد: روش اول این است که دوباره ذره را مقدار دهی اولیه نمود و روش دوم این است که ذره را مقداری دورتر از مکان بحرانی آن در فضای جستجو قرار داد. آزمایشات نشان داده که روش اول نتایجی بهتر از روش دوم دارد.

---

[58] New PSO

[59] Self organaized Criticality

۱.۳.۱.۷.۵ مدلهای دفعی Charged PSO

در [۴۲] گونه خاصی از الگوریتم ارائه شده است که از یک انرژی الکتروستاتیکی برای شارژ نمودن ذرات استفاده می‌نماید. یک مولفه به فرمول سرعت ذره اضافه شده است که از جذب ذرات به یکدیگر جلوگیری می‌نماید.

فصل دوم

پیاده‌سازی

در این بخش کدهای مربوط به نسخهٔ سراسری الگوریتم بهینه سازی دستهٔ ذرات که بـا اسـتفاده از نـرم افزار مطلب پیاده سازی شده است آورده شده:

```
    clear Pso;
    Pso.dimension=30;
    Pso.fitness_function_bound = 100;
    Pso.number = Pso.dimension * 5;
    Pso.max_itr=1000;
    Pso.Pbest_value = inf(Pso.number , 1);
    Pso.Pbest_position = zeros(Pso.number , Pso.dimension);
    Pso.X = Pso.fitness_function_bound * (rand(Pso.number ,
Pso.dimension));
    Pso.particle_value = zeros(Pso.number , 1);
    Pso.particle_value = fitness(Pso.X);
    Pso.Gbest_value = 0;
    [Pso.Gbest_value id] = min(Pso.particle_value);
    Pso.Gbest_position = Pso.X(id , :);
    Pso.Pbest_value = Pso.particle_value;
    Pso.Pbest_position = Pso.X;
    Pso.Gbest_per_each_itr = zeros(Pso.max_itr , 1);
    Pso.weight = 1;
    c1 = 2;
    c2 = 2;
    Pso.weight_min = 0.4;
    Pso.weight_max = 0.9;
    n=6;
    Pso.velocity = rand(Pso.number , Pso.dimension);
    for itr=1 : Pso.max_itr
       Pso.weight = (((Pso.max_itr - itr)/Pso.max_itr)*(Pso.weight_max-
Pso.weight_min)) + Pso.weight_min;
       for ii=1:Pso.number
            for jj=1:Pso.dimension
                if (Pso.X(ii,jj)>Pso.fitness_function_bound)
                   Pso.X(ii,jj)=Pso.fitness_function_bound ;
                   Pso.velocity(ii,jj) = 0;
                elseif (Pso.X(ii,jj)< (-1*Pso.fitness_function_bound))
                    Pso.X(ii,jj)= -1 * Pso.fitness_function_bound ;
                    Pso.velocity(ii,jj) = 0;
                end
            end
        end
        weight(itr) = Pso.weight;
        Gbest_temp = repmat(Pso.Gbest_position , Pso.number , 1);
        Pso.velocity = (Pso.weight .* Pso.velocity) + (c1*rand(Pso.number ,
Pso.dimension).*(Pso.Pbest_position - Pso.X)) + (c2*rand(Pso.number ,
Pso.dimension).*(Gbest_temp - Pso.X));
        Pso.X = Pso.X + Pso.velocity;
        Pso.particle_value = fitness(Pso.X , Pso.jjj);
        better = (Pso.particle_value < Pso.Pbest_value);
        for j = 1: Pso.number
            if(better(j))
               Pso.Pbest_value(j) = Pso.particle_value(j);
               Pso.Pbest_position(j , :) = Pso.X(j , :);
```

```
            end
        end
        [Pso.Gbest_value index] = min(Pso.Pbest_value);
        Pso.Gbest_position = Pso.Pbest_position(index , :);
        Pso.Gbest_per_each_itr(itr,1)=Pso.Gbest_value;
        [itr Pso.Gbest_per_each_itr(itr,1) jjj iii test] %#ok<NOPTS>
    end
```

شکل ۲-۱- کد PSO با استفاده از نرم افزار مطلب.

فصل سوم

جمع‌بندی و نتیجه‌گیری

در این پایان‌نامه به بررسی الگوریتم بهینه‌سازی دستهٔ ذرات و انواع متلف آن پرداختیم. در مدل‌های مختلف PSO سعی شده تا احتمال افتادن در بهینه‌های محلی کاهش یابد و توانایی جستجوی سراسری و جستجوی محلی الگوریتم بالا رود. یکی دیگر از معیارهایی که سعی شد تا افزایش یابد، سرعت همگرایی است. الگوریتم بهینه‌سازی دستهٔ ذرات به دلیل سادگی و کارایی بالا، یکی از محبوب‌ترین الگوریتم‌های بهینه‌سازی و هوش‌جمعی است.

مراجع

[1] - Z. Michalewicz and D. Fogel. How to Solve It: Modem Heuristics. Springer-Verlag، Berlin، 2000

[2] - P. Van Laarhoven and E. Aarts. SimulatedAnnealing: Theoty andApplications. Kluwer Academic Publishers، 1987

[3] - F. Glover. Tabu Search — Part I. URSA Journal on Computing، vol. 1، no. 3، pp. 190- 206، 1989.

[4] - Effects of Swarm Size on Cooperative Particle Swarm optimization S. Boettcher, A. G. Percus, Nature's way of optimizing. Artificial Intelligence 2000; 119; 275-286

[5] - C.W. Reynholds,Flocks. Herds and schools: A Distributed Behavioral Model. Computer Graphics,21(4):25-34,1987

[6] - Kennedy, J. and Eberhart, R. C., "Particle Swarm Optimization", Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ, pp. 1942-1948, 1995

[7] - E.O. Wilson, Sociobiology: The new Synthesis, Belknap Press 1975

[8] - R. Eberhart and Y. Shi. Comparison between Genetic Algorithms and Particle SwarmOptimization. In Proceedings of the Seventh Annual Conference on Evolutionary Programming، pp. 611–619. Springer-Verlag، 1998

[9] - R.C. Eberhart , P.K. Simpson, and R.W. Dobbins, Computational Intelligence PC Tools. Academic Press Professional first edition, 1996

[10] -J.Peng , Y. Chen,, and R.C. Eberhart,Battery Pack State of charge Estimator Design Using Computational Intelligence Approaches. In Proceedings of the annual Battery Conference on Applications and Advances, pages 173-177,2000

[11] -S. Naka ,T. Genji,T. Yura, and Fukuyama, Particle Distribution State Estimiation using Hybrid Particle Swarm Optimization . In IEEE power Engineering Society Winter Meeting, Volum2 , pages 815-820,January 2001

[12] -G. Venter , R.T. Haftka , and J. Sobieszczanski-Sobieski. Multidisciplinery Optimization of a Trasport Artificial Wing using Particle Swarm Optimization , In Ninth AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization 2002

[13] -M. Clerc Think Locally . Act Locally : The Way of Life of Cheap PSO an Adaptive PSO, Technique report , http://clerc.maurice.free.fr/pso/.2001

[14] -Y. Shi and R.C.Eberhart ,Fuzzy Adaptive Particle Swarm Optimization , In Proceedings of he IEEE Congeress on Evolutionary Computation ,volum 1,pages 101-106. IEEE press,May 2001

[15] -M. Clerc and J. Kennedy. The Particle Swarm: Explosion، Stability and Convergence in a Multi-Dimensional Complex Space. IEEE Transactions on Evolutionary Computation، vol. 6، pp. 58–73، ,2001

[16] -R. Brits, A.P. Engelbercht , and F. vanden Bergh, A Niching Particle Swarm Optimization , Techical Report, Department of Computer Sience University of Pritoria

[17] -P. N. Suganthan, "Particle swarm optimizer with neighborhood operator," in Proc. Congr. Evol. Comput., Washington, DC, 1999, pp. 1958–1962

[18] -A. Ratnaweera, S. Halgamuge, and H. Watson, "Particle Optimizer with Time varying Acceleration Coefficients . In Proceedings of the international Conference on Soft Computing and Inelligent Systems, pages 240-25, 2002

[19] -K.E. Parsopoulos, and M.N. Vrahatis, Recent Approach to Global Optimization Problems through Particle Swarm Optimization, Neural Copmuting, 1(2-3): 235-306,2002

[20] -C. Coello Coello and M. Lechuga. MOPSO: A Proposal for Multiple Objective Particle Swarm Optimization. In Congress on Evolutionary Computation، Piscataway، New Jersey، USA، vol. 2، pp. 1051–1056، IEEE Service Center، 2002

[21] -D. J. Watts and S.H. Strogatz. Collective Dynamics of 'Small-world' Networks Nature, 393(6684): 440-442, 1998

[22] -J. Kennedy ,Small Worlds and Mega-Minds : Effects of Neighborhood Topology onPsrticle Swarm Performance. In Proceeding s of The IEEECongress on Evolutionary Computation , volum 3, pages 1931-1938 , July 1999

[23] -S. Janson and M Middendorf. A Hierarchical Particle Swarm Optimization , In Proceedings of the IEEE Congress on Evolutionary Computation , volume 2 . pages 770-776, December 2003

[24] -P. Suganthan. Particle Swann Optimizer with Neighborhood Optimizer. In Proceedings of the Congress on Evolutionary Computation، PP. 1958–1962، 1999

[25] -R.Mendes , J Kennedy , and J Neves. Watch they Neighbor or How the Swarm can Learn from its Environment . In Proceedings of the IEEE Swarm Intelligence Symposium

[26] -T. Ray , K.M. Liew, and Saini An Intelligent Information Strategy within a Swarm for Unconstrained and Constrained Optimization problem , Soft Computing – A Fusion of Foundations, Methodologies and Applications 6(1), 2002

[27] -Y. Fukuyama and H. Yoshida. A Particle Swarm Optimization for Reactive Power and Voltage Control in Electric Power Systems. In Proceedings of the IEEE Conress on Evolutionary Computation، Seoul، S. Korea، pp. 87–93، , 2001

[28] -N. Higashi and H. Iba. Particle Swami Optimization with Gaussian Mutation. In Proceedings of the IEEE Swarm Intelligence Symposium 2003 (SIS 2003)، Indianapolis، Indiana، USA. pp. 72–79، 2003

[29] -A. Stacey , M. Jancic, and I Grundy, Particle Swarm Optimization with mutation , In Proceedings of the IEEE Congress on Evolutionary Computation, volume 2, pages 1425-1430, 2003

[30] -T-O. Ting , M.V. C. Rao , C.K. Loo, and S-S. Ngu. A New Class of Operators to Accelerate Particle Swarm Optimization . InProceedings of the IEEE Congress on Evolutionary Computation, volum 4, pages 2406-2410, December 2003

[31] -J. Fieldsend and S. Singh. A Multi-objective Algorithm based upon Particle Swarm Optimization، an Efficient Data Stmcture and Turbulence. In The 2002 UK Workshop on Computational Intelligence، UK، pp. 34–44، 2002.

[32] -V. Miranda and N. Fonseca, "New evolutionary particle swarm algorithm (EPSO) applied to voltage/VAR control," in Proc. 14th Power Syst. Comput. onf., Seville, Spain, 2002. [Online]. Available: http://www.pscc02.org/papers/s21pos.pdf.

[33] -Liu, H., Abraham, A. and Zhang, W. (2007) 'A fuzzy adaptive turbulent particle swarm optimisation', Int. J. Innovative Computing and Applications, Vol. 1, No. 1, pp.39–47

[34] -F. Van den Bergh and A.P. Engelbrecht. Effects of Swarm Size on Cooperative Particle Swarm Optimizers. In Proceedings of the Genetic andEvolutionary Computation Conference، San Francisco، USA، pp. 892–899، 2001.

[35] -J. Riget and J.S. Vesterstom, A Diversity-Guided Particle Swarm Optimizer The The ARPSO ,Thechnical report ,Department of Computer Sience ,University of Aarhus,2002

[36] -G. Theraulaz, S Goss , J,Gervet, and J-L Deneubourg. Task Differentation in Polists Wasp Colonies : A model for Self-Organaizing of Robots. In J=A Meyer and S.W. Wilson, editors , Proceedings of the First International Conference on Simulation of Adaptive Behavior: From Animal to Animate, pages 346-355, MIT Press,1991

[37] -S.H.Clearwater, T.Hogg, and B.A. Huberman,"Cooperative problem solving," in Camputation : The Micro and Macro View, Singapore: World Sceintific,1992,pp.33-70

[38] -J.S. Vesterstrom , J. Giget, and T.Krink,Division of Labor in Particle Swarm Optimization . In Proceedings of the IEEE Congress on Evolutionary Computation , pages 1570-1575 IEEE press,2002

[39] -J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Particle swarm optimization algorithms with novel learning strategies," in Proc. Int. Conf. Systems, Man, Cybernetics, The Netherlands, Oct. 2004. [Online]. Available: http://www.ntu.edu.sg/home/EPNSugan

[40] -B.B. Thompson, R.J. Marks,M.A. El-Shorkawi,W.J. Fox ,and R.T. Inversion of Neural Network Underwater Acoustic Model for Estimation of Bottom Parameters using Modified Particle Swarm Optimizer In Proceedings of the International Joint Conference on Nural Network ,page 1306,2003

[41] -A New Particle Swarm Optimization Technique Chunming Yang and Dan Simon Electrical and Computer Engineering Department Cleveland State University Cleveland، Ohio

[42] -T. M. Blackwell and P. J. Bentley, "Don't push me! Collision-avoiding swarms," in Proc. IEEE Congr. Evol. Comput., Honolulu, HI, 2002, pp. 1691–1696