

پیوست ب

روش‌های بهینه‌سازی

در این بخش ابتدا رویکردهای الگوریتم ژنتیک و بهینه‌سازی ازدحام ذرات معرفی می‌شود. سپس با توجه به اینکه در اکثر مسائل مهندسی بیش از یک تابع هدف برای بهینه‌سازی وجود دارد نکاتی در ارتباط با بهینه‌سازی چندهدفه بیان می‌گردد.

ب-۱ الگوریتم ژنتیک

الگوریتم ژنتیک یکی از توسعه یافته‌ترین روش‌های تکاملی و به طور کلی یکی از قدرتمندترین رویکردهای بهینه‌سازی می‌باشد و در نتیجه در مسائل گوناگون بهینه‌سازی همچون بهبود عملکرد تجهیزات صنعتی بسیار پرکاربرد می‌باشد. ایده‌ی اولیه‌ی الگوریتم ژنتیک توسط **جان هالند** (۱۹۷۵) با اقتباس از نظریه تکامل ^۱ **چارلز داروین** (۱۸۵۹) ارائه شد. بر اساس این نظریه، تکامل گونه‌های زیستی در طول زمان بر اساس انتخاب طبیعی ^۲ به نحوی است که گونه‌های ضعیف‌تر، به تدریج جای خود را به گونه‌های دارای برازندگی ^۳ بیشتر با تطبیق‌پذیری ^۴ بالاتر نسبت به طبیعت می‌دهند که این کار از طریق انتقال صفات برتر از نسلی به نسل دیگر صورت می‌گیرد. عامل اصلی انتقال صفات زیستی در بین گونه‌های زنده، کروموزوم‌ها ^۵ و ژن‌ها ^۶ هستند و دانشی که به بررسی نحوه انتقال ژن‌ها در نسل‌های مختلف می‌پردازد علم ژنتیک می‌باشد.

مفاهیم الگوریتم ژنتیک نیز از ژنتیک طبیعی اقتباس شده است. در این رویکرد هر طراحی پیشنهادی

¹Evolution

²Natural selection

³Fitness

⁴Adaption

⁵Chromosome

⁶Gene

برای سیستم به عنوان یک فرد ^۷ در نظر گرفته می‌شود که به مرور زمان بهبود داده می‌شود. همچنین از آن‌جا که فرآیند انتقال ژن در طبیعت در بستر تنوع صفات رخ می‌دهد لازم است در هر مرحله بجای یک عضو، یک مجموعه از افراد به نام جمعیت ^۸ در نظر گرفته شود. بدین ترتیب هر مرحله از بهینه‌سازی نیز معرف یک نسل ^۹ می‌باشد. در تطابق با مفاهیم ژنتیک طبیعی، برازندگی هر فرد معادل میزان مطلوبیت آن است که می‌تواند متناسب با عملکرد سیستم در آن طراحی یا به نسبت عکس تابع هدف کمینه‌سازی فرض شود. فرآیند بهینه‌سازی با استفاده از این رویکرد بدین صورت است که در ابتدا یک مجموعه طراحی اولیه برای سیستم به عنوان نسل اولیه در نظر گرفته می‌شود. سپس تلاش می‌شود با الگوبرداری از مکانیزم‌های احتمالی انتقال ژن در موجودات زنده همچون جهش ^{۱۰} و همگذاری ^{۱۱}، عواملی که احتمالاً منجر به افزایش برازندگی یک عضو است از یک یا دو فرد به عنوان والدین به فرزندان جدید انتقال یابد. برای این رسیدن به این منظور نیز فرآیند انتخاب ^{۱۲} والد/والدین بر مبنای فرضیه انتخاب طبیعی و به صورت احتمالاتی و با احتمالی متناسب با برازندگی افراد صورت می‌گیرد. بدین ترتیب برازندگی جمعیت جدید به طور کلی افزایش می‌یابد تا به طراحی بهینه برسد. الگوریتم ژنتیکی که در ابتدا توسط **جان هالند** (۱۹۷۵) معرفی گردید بسیار ساده بود. این الگوریتم توسط محققین بسیاری به شکل‌های مختلف و به منظور افزایش سرعت همگرایی و قدرت جستجوی سراسری در کاربردهای گوناگون ریاضی و مهندسی توسعه یافته است (**گلدبرگ و سمتانی**، ۱۹۸۶)؛ (**سرنیواس و دب**، ۱۹۹۴)؛ (**دب و دیگران**، ۲۰۰۷)؛ (**دب و دب**، ۲۰۱۴)؛ (**ساری و دیگران**، ۲۰۲۰). روشی که در این رساله مورد استفاده قرار گرفته الگوریتم ژنتیک پیوسته ^{۱۳} است که متغیرهای طراحی را پیوسته و به صورت مقدار حقیقی ^{۱۴} در نظر می‌گیرد. این روش در مقابل رویکرد الگوریتم ژنتیک دودویی ^{۱۵} که در ابتدا توسط **جان هالند** (۱۹۷۵) و برای متغیرهای طراحی گسسته پیشنهاد گردید قرار دارد. در ادامه جزئیات روش مورد استفاده در رساله ارائه گردیده است.

⁷Individual⁸Population⁹Generation¹⁰Mutation¹¹Crossover¹²Selection¹³Continuous genetic algorithm¹⁴Real value¹⁵Binary genetic algorithm

ب-۱-۱ تشکیل جمعیت اولیه

اولین قدم در الگوریتم ژنتیک تولید جمعیت اولیه $G^{(1)}$ یا نسل اول است. همان‌طور که پیش‌تر اشاره شد هر جمعیت از مجموعه‌ای از افراد یا کروموزوم‌ها^{۱۶} ایجاد شده است به گونه‌ای که هر کروموزوم معرف یک نقطه در فضای طراحی $\mathbb{D}$ بوده و از رشته‌ای^{۱۷} از ژن‌ها تشکیل شده است. در الگوریتم ژنتیک مقدار حقیقی، هر ژن برابر با مقدار یک متغیر طراحی می‌باشد. در نتیجه طول هر کروموزوم برابر با تعداد متغیرهای طراحی D است. از این رو در هر نسل، اعضای جمعیت به صورت زیر می‌باشند:

$$G^{(g)} = \{x_1^{(g)}, x_2^{(g)}, \dots, x_i^{(g)}, \dots, x_N^{(g)}\}, \quad (ب-۱)$$

$$x_i^{(g)} = \{x_{i,1}^{(g)}, x_{i,2}^{(g)}, \dots, x_{i,j}^{(g)}, \dots, x_{i,D}^{(g)}\}, \quad \mathcal{N}(x_i^{(g)}) \leq 0,$$

که در آن g شماره نسل، N تعداد اعضای جمعیت $x_i^{(g)}$ کروموزوم یا فرد i ام و $x_{i,j}^{(g)}$ ژن j ام کروموزوم i می‌باشد. بدین ترتیب برای ایجاد نسل اول لازم است تعداد N کروموزوم یا طراحی مجاز پیشنهادی به شکل $G^{(1)} = \{x_1^{(1)}, x_2^{(1)}, \dots, x_i^{(1)}, \dots, x_N^{(1)}\}$ تولید گردد (رایت، ۱۹۹۱). این کار با هر یک از روش‌های نمونه‌برداری که در بخش ۳-۳-۴ توضیح داده شد می‌تواند صورت گیرد. در این جا نیز طبیعتاً هرچه نمونه‌های اولیه، فضای طراحی را بهتر پوشش دهند فرآیند بهینه‌سازی احتمالاً از سرعت همگرایی بالاتری برخوردار است. از این رو در این رساله برای تولید جمعیت اولیه از روش نمونه‌برداری سوپول استفاده شده است. همچنین کروموزوم‌های ایجاد شده باید در فضای مجاز طراحی قرار داشته باشند یا به عبارتی $\mathcal{N}(x_i^{(g)}) \leq 0$. بنابراین پس از تولید هر فرد، قیود حاکم بر مسئله ارزیابی می‌شود و در صورتیکه طراحی پیشنهادی، قیود را ارضا نکند کروموزوم تولید شده قابل قبول نمی‌باشد. البته لازم به ذکر است در برخی مسائل پیچیده، بررسی قیود در ابتدای تشکیل جمعیت امکان‌پذیر نیست و به عنوان مثال تا رفتار سیستم شبیه‌سازی نشود نمی‌توان در این باره اظهار نظر نمود. در این‌گونه مسائل معمولاً قیود در هنگام تشکیل جمعیت ارزیابی نشده و پس از بررسی مدل سیستم، قیودی که ارضا نشده‌اند به عنوان یک جریمه^{۱۸} در تابع هدف در نظر گرفته می‌شوند تا این نقاط به صورت خودکار در فرآیند بهینه‌سازی حذف شوند. در نهایت فرآیند تشکیل جمعیت تا تولید N کروموزوم ادامه می‌یابد. همچنین بهتر است اگر حدس‌های اولیه‌ای همچون طراحی‌های پایه برای سیستم وجود دارد آن‌ها نیز در جمعیت اولیه در نظر گرفته شوند تا فرآیند بهینه‌سازی تسریع گردد. با این وجود توجه

¹⁶Chromosome

¹⁷String

¹⁸Penalty

به این نکته ضروری است که حدس‌های اولیه نباید بخش عمده‌ی جمعیت اولیه را تشکیل دهند تا امکان جستجوی تمام فضای طراحی وجود داشته باشد.

ب-۱-۲ برازندگی

پس از تشکیل جمعیت نسل g لازم است برازندگی هر کروموزوم محاسبه گردد. همان طور که پیش‌تر بیان گردید برازندگی معیاری از مطلوبیت هر کروموزوم است که می‌تواند معرف میزان بهینه بودن نسبی هر طراحی باشد. از این رو ابتدا باید مقدار تابع هدف برای هر کروموزوم به شکل $y_i^{(g)} = \mathcal{J}(x_i^{(g)})$ محاسبه گردد و سپس با تعریف یک عملگر مناسب، به هر تابع هدف یک برازندگی مثبت $f_i^{(g)} = \mathcal{F}(y_i^{(g)})$ متناظر گردد. به عنوان مثال اگر هدف از بهینه‌سازی، کمینه‌سازی باشد، عملگر برازندگی می‌تواند به صورت

$$\mathcal{F}(y) = \sup\{y\} - y \quad (\text{ب-۲})$$

تعریف گردد که در آن $\sup$ نمایانگر کوچک‌ترین کران بالا ^{۱۹} یا زیرینه ^{۲۰} است. البته لازم به ذکر است در رابطه (ب-۲) بجای کوچک‌ترین کران بالا از هر یک از کران‌های بالا نیز می‌توان استفاده کرد. یکی دیگر از تعریف‌های برازندگی می‌تواند بدین صورت باشد که مقدار تابع هدف برای تمام افراد مورد نظر محاسبه گردد. سپس اعضا به ترتیب از مقدار کمینه (بهترین) به سمت بیشینه (بدترین) مرتب گردیده و به هر فرد یک اندیس بر حسب جایگاه آن نسبت داده شود. در اینصورت بهترین مقدار جایگاه اول و بدترین مقدار جایگاه آخر قرار دارد. حال مقدار برازندگی هر فرد، برابر تعداد کل افراد منهای جایگاه آن فرد به علاوه یک (جهت جلوگیری از تولید برازندگی صفر) تعریف می‌شود.

ب-۱-۳ تولید نسل جدید

همان‌طور که پیش‌تر اشاره گردید، رویکرد بهینه‌سازی در الگوریتم ژنتیک بر مبنای تولید نسل‌های جدید با صفات برتر است. برای ایجاد هر نسل جدید لازم است در ابتدا به تعداد مورد نیاز فرزند جدید تولید شود. این فرآیند با استفاده از سه عملگر انتخاب، هم‌گذاری و جهش صورت می‌گیرد. عملگر انتخاب، برای انتخاب والدین و همچنین اعضای جمعیت جدید است در حالی که دو عملگر هم‌گذاری و جهش برای تولید فرزندان جدید با استفاده از والدین انتخاب شده به کار می‌رود. در مکانیزم هم‌گذاری دو فرزند با استفاده از

¹⁹Least upper bound

²⁰Supremum

دو والد و در مکانیزم جهش یک فرزند جدید به کمک یک والد تولید می‌شود. بنابراین در هر مرحله اولین قدم این است که مکانیزم تولید فرزند/فرزندان انتخاب شود. این کار نیز به صورت احتمالی صورت می‌گیرد بدین ترتیب که احتمال تولید دو فرزند با استفاده از مکانیزم همگذاری برابر p_c و احتمال تولید یک فرزند با استفاده از مکانیزم جهش برابر p_m است. لازم به ذکر است که این دو فرآیند می‌توانند مکمل هم نبوده و مستقل از یکدیگر باشند. **دب و آگراوال (۱۹۹۹)** مقادیر زیر را برای احتمال این دو مکانیزم پیشنهاد دادند:

$$p_c = 0.9, \quad (b-3)$$

$$p_m = \frac{1}{D} + C \cdot g \left(1 - \frac{1}{D}\right),$$

که در آن، D تعداد متغیرهای طراحی، g شماره نسل و C یک مقدار ثابت است. ثابت C به گونه‌ای انتخاب می‌شود که تا انتهای فرآیند بهینه‌سازی احتمال جهش بیش از مقدار ۱ نباشد، به عنوان مثال اگر بودجه‌ی محاسباتی کل به اندازه‌ای باشد که تعداد کل نسل‌ها از مقدار $g_{\max}$ تجاوز نکند، این ثابت را می‌توان به صورت $C = 1/g_{\max}$ در نظر گرفت. پس از مشخص شدن مکانیزم تولید فرزندان و با کمک مکانیزم انتخاب، تعداد والد مورد نیاز مشخص شده و فرزندان بر اساس آن‌چه در ادامه توضیح داده خواهد شد تولید می‌شوند. با توجه به استقلال احتمال دو مکانیزم مذکور، در هر مرحله از تولید فرزندان ممکن است هیچ فرزندی تولید نشود، یک فرزند با استفاده از جهش، دو فرزند با استفاده از همگذاری و یا سه فرزند با استفاده از هر دو مکانیزم تولید شود. در نتیجه مراحل تولید فرزندان تا تولید N فرزند جدید ادامه می‌یابد. در این مرحله روش‌های متعددی برای تشکیل نسل جدید بر اساس فرزندان تولید شده پیشنهاد گردیده است. بر اساس پیشنهاد **شفر (۱۹۸۶)**، برای ایجاد نسل جدید، مجموعه‌ای متشکل از N عضو جمعیت قبل و N فرزند جدید C ایجاد می‌گردد و با استفاده از مکانیزم انتخاب تعداد N فرد به عنوان اعضای نسل جدید انتخاب می‌شوند.

ب-۱-۴ انتخاب

عملگر انتخاب برای انتخاب والدین در تولید فرزندان و همچنین انتخاب اعضای نسل جدید به کار می‌رود. بر مبنای رفتار طبیعت، پیش‌بینی می‌شود افراد برازنده‌تر شانس بیشتری برای بقا داشته باشند. همچنین فرزندانی که از والدین برازنده‌تر متولد می‌شوند خود نیز برازنده‌تر باشند. این فرآیند در الگوریتم ژنتیک نیز

شبیه‌سازی شده و به کروموزوم‌های برازنده‌تر شانس بیشتری برای تولید مثل و بقا می‌دهد. از این رو تمامی انتخاب‌ها در الگوریتم ژنتیک به صورت احتمالاتی و با شانس متناسب با میزان برازندگی نسبی هر فرد از بین کل افراد جمعیت انجام می‌شود. این کار می‌تواند به کمک گردونه‌ی شانس^{۲۱} صورت پذیرد. فرض کنید نیاز به انتخاب تعدادی عضو از مجموعه P باشد. در این صورت عملگر انتخاب M عضو از مجموعه مذکور به صورت $\mathcal{S}_M\{P\}$ نشان داده می‌شود. حال یک گردونه شانس متشکل از $|P|$ قطاع (برابر با تعداد اعضای P) به گونه‌ای ایجاد می‌شود که هر قطاع متناظر با یکی از افراد جمعیت بوده و زاویه قطاع i ام متناظر با عضو x_i که دارای برازندگی f_i می‌باشد برابر با $f_i / \sum_{j=1}^{|P|} f_j$ می‌باشد. حال با هر بار چرخاندن گردونه که متناظر با ایجاد یک عدد تصادفی در بازه $[0, 2\pi)$ است یک عضو انتخاب می‌شود. این عمل تا انتخاب M عضو متمایز ادامه پیدا می‌کند. برای انتخاب والدین نکته‌ای که وجود دارد این است که بر اساس توصیه شفر (۱۹۸۶) تمامی اعضا شایستگی زاد و ولد ندارند. روندی که در این رساله استفاده شده بدین ترتیب است که ابتدا یک خزانه‌ی جفت‌گیری^{۲۲} (M) متشکل از افرادی که امکان والد بودن دارند تشکیل می‌شود و سپس برای هر زاد و ولد، والدین از بین اعضای این مجموعه انتخاب می‌شوند. بر این اساس اعضای خزانه‌ی جفت‌گیری به تعداد $N/2$ عضو با استفاده از مکانیزم انتخاب و از بین کل افراد جمعیت تعیین می‌گردند:

$$\mathcal{M} = \mathcal{S}_{N/2}\{G\}. \quad (\text{ب-۴})$$

ب-۱-۵ همگذاری

عملگر همگذاری یکی از عملگرهای پرکاربرد در الگوریتم ژنتیک می‌باشد که با توجه به احتمال بالای وقوع آن نقش تعیین‌کننده‌ای در فرآیند بهینه‌سازی دارد. همگذاری به نوعی پیاده‌سازی ایده‌ی جستجوی بر مبنای استخراج می‌باشد. بدین صورت که این عملگر، اطلاعات بین رشته‌ها را به طور تصادفی با یکدیگر مبادله می‌کند. در الگوریتم ژنتیک دودویی، ساده‌ترین نوع همگذاری که همگذاری تک نقطه‌ای^{۲۳} نام دارد بدین شکل است که یک نقطه روی دو کروموزوم والد انتخاب شده و ژن‌های بعد از آن دو نقطه با یکدیگر معاوضه می‌شوند و بدین ترتیب دو فرزند جدید متولد می‌گردد. در الگوریتم ژنتیک پیوسته ایده‌های گوناگونی برای عملگر همگذاری توسعه یافته است. یکی از پرکاربردترین رویکردها در این زمینه همگذاری دودویی شبیه‌سازی

²¹Roulette wheel

²²Mating pool

²³Single-point crossover

شده ^{۲۴} است که توسط **دب و آگراوال (۱۹۹۵)**؛ **کومار و دب (۱۹۹۵)**؛ **دب و بیر (۲۰۰۱)** ارائه گردید. همان‌طوری که از اسم این مکانیزم مشخص است، در طراحی این عملکرد سعی بر این بوده که اصول عملکرد همگذری تک نقطه‌ای پیاده‌سازی گردد. در ادامه روند تولید دو فرزند $x_{c_1}^{(g)}$ و $x_{c_2}^{(g)}$ در نسل شماره g از دو والد $x_{p_1}^{(g)}$ و $x_{p_2}^{(g)}$ در این نسل با استفاده از این عملکرد توضیح داده شده است. در این رویکرد برای ژن i اُم که متناظر با i اُمین متغیر طراحی x_i است ضریب انتشار ^{۲۵} θ به صورت نسبت اختلاف مقادیر مربوطه در فرزندان به والدین تعریف می‌گردد:

$$\theta = \left| \frac{x_{c_2,i}^{(g)} - x_{c_1,i}^{(g)}}{x_{p_2,i}^{(g)} - x_{p_1,i}^{(g)}} \right|. \quad (\text{ب-۵})$$

دب و آگراوال (۱۹۹۵) نشان دادند برای اینکه قدرت جستجوی ^{۲۶} این عملکرد با همگذری تک نقطه‌ای دودویی برابر باشد لازم است تابع توزیع احتمال θ به شکل زیر در نظر گرفته شود:

$$f_{\theta} = \begin{cases} \frac{\rho_c+1}{4} \theta^{\rho_c} & \text{if } \theta \leq 1; \\ \frac{\rho_c+1}{4} \theta^{-(\rho_c+2)} & \text{otherwise} \end{cases}, \quad (\text{ب-۶})$$

که در آن ρ_c شاخص توزیع همگذری است و هرچه مقدار آن بیشتر باشد، فرزندان تولید شده شباهت بیشتری به والدین خود دارند. در این رساله از $\rho_c = 20$ استفاده شده است. **دب و بیر (۲۰۰۱)** برای سهولت در

پیاده‌سازی این رویکرد ابتدا تابع توزیع تجمعی θ را به شکل

$$\Phi = F_{\theta} = \begin{cases} \frac{1}{4} \theta^{\rho_c+1} & \text{if } \theta \leq 1; \\ 1 - \frac{1}{4} \theta^{-(\rho_c+1)} & \text{otherwise} \end{cases}, \quad (\text{ب-۷})$$

محاسبه نمودند و سپس رابطه فوق را به شکل زیر بازنویسی کردند:

$$\theta = \begin{cases} (4\Phi)^{\frac{1}{\rho_c+1}} & \text{if } \Phi \leq 0.5; \\ \left(\frac{1}{4(1-\Phi)} \right)^{\frac{1}{\rho_c+1}} & \text{otherwise} \end{cases}, \quad (\text{ب-۸})$$

در اینصورت برای محاسبه ضریب انتشار θ کافی است Φ یک عدد تصادفی در بازه $[0, 1]$ اختیار شده و سپس مقدار θ با استفاده از رابطه **(ب-۸)** بدست می‌آید. در نهایت با فرض اینکه میانگین مقدار هر متغیر طراحی در والدین و فرزندان یکسان باشد یا به عبارتی:

$$x_{c_1,i}^{(g)} + x_{c_2,i}^{(g)} = x_{p_1,i}^{(g)} + x_{p_2,i}^{(g)}, \quad (\text{ب-۹})$$

^{۲۴}Simulated binary crossover (SBC)

^{۲۵}Spread factor

^{۲۶}Search power

می‌توان ژن i ام فرزندان را به شکل زیر تولید کرد:

$$x_{c_1,i}^{(g)} = \frac{1}{2} \left[(1 + \theta)x_{p_1,i}^{(g)} + (1 - \theta)x_{p_2,i}^{(g)} \right], \quad (\text{ب-۱۰})$$

$$x_{c_2,i}^{(g)} = \frac{1}{2} \left[(1 - \theta)x_{p_1,i}^{(g)} + (1 + \theta)x_{p_2,i}^{(g)} \right].$$

بدین ترتیب هر فرزند یک والد اصلی دارد که به آن شبیه‌تر بوده و یک والد فرعی دارد که برای فرزند دیگر والد اصلی به حساب می‌آید.

ب-۱-۶ جهش

می‌توان گفت عملگرهای انتخاب و هم‌گذاری وظیفه پیاده‌سازی تولید مثل فرزندان برتر از والدین اصلح و بقای اعضای برازنده‌تر را به عهده دارند. با این حال همواره ممکن است یک سری صفات مثبت به صورت بالقوه وجود داشته باشند که در هیچ از اعضای جمعیت حاضر نباشد یا در اثر فرآیندهای انتخاب و هم‌گذاری به صورت تصادفی از جمعیت حذف شده باشند. از این رو اهمیت اکتشاف به منظور جستجوی ژن‌های خوب روشن می‌شود. عملگری که این وظیفه را در الگوریتم ژنتیک پیاده‌سازی می‌کند جهش نام دارد. در الگوریتم ژنتیک دودویی این کار با انتخاب تصادفی یک ژن و تغییر آن از ۰ به ۱ و یا بالعکس صورت می‌گیرد. در الگوریتم‌های ژنتیک پیوسته رویکردهای متفاوتی برای رسیدن به این منظور پیشنهاد شده است (دب و گویال، ۱۹۹۶؛ دب و آگراوال، ۱۹۹۹؛ دب و دب، ۲۰۱۴). رویکردی که در این رساله مورد استفاده قرار گرفته است روش جهش چندجمله‌ای^{۲۷} می‌باشد. دب و گویال (۱۹۹۶) برای اعمال جهش از تعریف ضریب اختلال^{۲۸} δ برای اختلاف بین ژن i ام فرزند $x_{c,i}^{(g)}$ و والد $x_{p,i}^{(g)}$ استفاده کردند:

$$\delta = \frac{x_{c,i}^{(g)} - x_{p,i}^{(g)}}{\delta_{\max}}, \quad (\text{ب-۱۱})$$

که در آن $\delta_{\max}$ حداکثر مقدار اختلال مجاز بوده و بنابر توصیه دب و دب (۲۰۱۴) به صورت زیر انتخاب می‌شود:

$$\delta_{\max} = \begin{cases} x_{p,i}^{(g)} - \inf\{x_i\} & \text{if } x_{c,i}^{(g)} \leq x_{p,i}^{(g)}; \\ \sup\{x_i\} - x_{p,i}^{(g)} & \text{otherwise} \end{cases}. \quad (\text{ب-۱۲})$$

^{۲۷}Polynomial mutation

^{۲۸}Perturbance factor

در رابطه فوق، $\inf\{x_i\}$ و $\sup\{x_i\}$ به ترتیب بزرگترین کران پایین^{۲۹} (زیرینه^{۳۰}) و کوچکترین کران بالا (زیرینه) دامنه‌ی تعریف متغیر طراحی x_i می‌باشند. همانند عملگر همگذاری، **دب و آگراوال** (۱۹۹۹) یک توزیع احتمال برای ضریب اختلال در نظر گرفتند. این توزیع به شکل

$$f_{\delta} = \frac{1}{\gamma}(\rho_m + 1)(1 - |\delta|)^{\rho_m} \quad (\text{ب-۱۳})$$

بوده و ρ_m در آن شاخص توزیع جهش نام دارد که بر اساس پیشنهاد **دب و دب** (۲۰۱۴) در بازه $[0, 100]$ انتخاب می‌شود. در اینجا نیز برای پیاده‌سازی جهش لازم است در ابتدا تابع توزیع تجمعی δ به شکل

$$\Phi = F_{\delta} = \begin{cases} \frac{1}{\gamma}(1 + \delta)^{\rho_m + 1} & \text{if } \delta \leq 0; \\ 1 - \frac{1}{\gamma}(1 - \delta)^{\rho_m + 1} & \text{otherwise} \end{cases} \quad (\text{ب-۱۴})$$

در نظر گرفته شده و برای محاسبه ضریب اختلال مقدار Φ به صورت تصادفی در بازه $[0, 1]$ انتخاب گردد. سپس با استفاده از رابطه زیر مقدار δ محاسبه می‌شود:

$$\delta = \begin{cases} (\gamma\Phi)^{\frac{1}{\rho_m + 1}} - 1 & \text{if } \Phi \leq 0.5; \\ 1 - (\gamma(1 - \Phi))^{\frac{1}{\rho_m + 1}} & \text{otherwise} \end{cases} \quad (\text{ب-۱۵})$$

بر این اساس ژن i اُم فرزند جدید با استفاده از رابطه **(ب-۱۱)** به صورت زیر بدست می‌آید:

$$x_{\epsilon, i}^{(g)} = x_{p, i}^{(g)} + \delta \cdot \delta_{\max}. \quad (\text{ب-۱۶})$$

ب-۲ بهینه‌سازی ازدحام ذرات

بهینه‌سازی ازدحام ذرات یکی از روش‌های بهینه‌سازی قدرتمند و پرکاربرد در مسائل مهندسی می‌باشد که با الگوبرداری از رفتار اجتماعات زیستی توسعه یافته است. **کندی و ابرهات** (۱۹۹۵) با تلاش در جهت توسعه نوعی از هوش مصنوعی^{۳۱} بر مبنای شبیه‌سازی روابط اجتماعی موجودات زنده، موفق شدند این رویکرد بهینه‌سازی را با استفاده از پیاده‌سازی رفتار پرندگان در یافتن دانه توسعه دهند. بر این اساس حرکت پرندگان در هر لحظه به سمت غذا با توجه به تجربیات شخصی و گروهی صورت می‌گیرد. در این الگوریتم نیز فرآیند بهینه‌سازی به کمک یک مجموعه از ذرات و پیگیری حرکت آن‌ها در طول زمان به سمت موقعیت هدف صورت می‌گیرد. از این رو برای هر ذره دو مشخصه موقعیت و سرعت حرکت تعریف می‌گردد. به

²⁹Greatest lower bound

³⁰Infimum

³¹Artificial intelligence

علاوه هر ذره معرف مقدار تابع هدف بهینه‌سازی در موقعیتی از فضا که در آن قرار گرفته است بوده و در نتیجه موقعیت هدف، معادل نقطه بهینه می‌باشد. بر این اساس، ابتدا یک مجموعه جمعیت اولیه ذرات در فضای جستجو یا متغیرهای طراحی تولید می‌شود. سپس سرعت حرکت آن‌ها در هر لحظه بدست آمده و موقعیت جدید آن‌ها بدست می‌آید. جزئیات این روش در ادامه توضیح داده شده است.

ب-۲-۱ تشکیل اجتماع

در بهینه‌سازی ازدحام ذرات نیز همچون الگوریتم ژنتیک، اولین مرحله، تشکیل اجتماع ذرات می‌باشد. از آن‌جا که هر ذره با دو مشخصه موقعیت و سرعت معرفی می‌شود، اجتماع ذرات در هر مرحله از حرکت t به صورت زیر تعریف می‌شود:

$$\mathcal{S}^{(t)} = \{(x_1^{(t)}; v_1^{(t)}), (x_2^{(t)}; v_2^{(t)}), \dots, (x_i^{(t)}; v_i^{(t)}), \dots, (x_N^{(t)}; v_N^{(t)})\},$$

$$x_i^{(t)} = \{x_{i,1}^{(t)}, x_{i,2}^{(t)}, \dots, x_{i,j}^{(t)}, \dots, x_{i,D}^{(t)}\}, \quad \mathcal{N}(x_i^{(t)}) \leq *, \quad (\text{ب-۱۷})$$

$$v_i^{(t)} = \{v_{i,1}^{(t)}, v_{i,2}^{(t)}, \dots, v_{i,j}^{(t)}, \dots, v_{i,D}^{(t)}\},$$

که در آن t شماره مرحله حرکت گروه، N تعداد اعضای جمعیت $x_i^{(t)}$ موقعیت ذره i ام و معرف یک نقطه در فضای طراحی یا فضای جستجو، $v_i^{(t)}$ سرعت این ذره و $x_{i,j}^{(t)}$ و $v_{i,j}^{(t)}$ به ترتیب مولفه j ام موقعیت و سرعت ذره i می‌باشند. در اینصورت برای تشکیل اجتماع ذرات در لحظه اول ضروری است موقعیت و سرعت ذرات در این لحظه مشخص شوند. موقعیت ذرات با استفاده از تکنیک نمونه‌برداری سوبول به صورت شبه تصادفی در فضای طراحی مشخص می‌شود. طبیعی است ذرات تولید شده باید قیود حاکم بر مسئله را ارضا کنند و در نتیجه تولید نمونه‌ها تا دستیابی به N نقطه در فضای مجاز طراحی ادامه می‌یابد. در مرحله‌ی بعد باید سرعت اولیه ذرات مشخص شود. حرکت تمامی ذرات در این رویکرد از حالت سکون در نظر گرفته می‌شود و در نتیجه $v_i^{(1)} = \{0\}^D$. البته در برخی از روش‌های بهینه‌سازی ازدحام ذرات سرعت اولیه آن‌ها نیز برابر با یک مقدار تصادفی در نظر گرفته می‌شود (وانگ و دیگران، ۲۰۱۸).

ب-۲-۲ حرکت ذرات

همان‌طور که پیش‌تر بیان گردید دستیابی به نقطه بهینه از طریق پیگیری حرکت ذرات حاصل می‌شود. حرکت ذرات به معنی جابجایی آن‌ها از موقعیت فعلی و متناسب با سرعت ذرات می‌باشد. سرعت هر ذره در هر لحظه که بیان‌گر میزان تغییر موقعیت آن ذره می‌باشد با استفاده تجربیات شخصی ذره و اجتماع بدست می‌آید. با

این حال اینرسی ذره باعث می‌شود تغییر جهت ناگهانی امکان‌پذیر نباشد. از این رو سرعت ذره برای لحظه بعد بر اساس برآیندی از سرعت ذره در این لحظه، حرکت به سمت بهترین نقطه‌ای که در گذشته در آن حضور داشته و حرکت به سمت یک یا چند ذره از بهترین ذرات موجود در اجتماع صورت می‌گیرد. البته لازم به ذکر است این برآیند برای تمام ذرات یکسان نبوده و هر ذره‌ای در هر لحظه‌ای ممکن است به شکل متفاوتی از دانش فردی و گروهی استفاده کند. بدین ترتیب مولفه j ام سرعت ذره در لحظه $t + 1$ به صورت زیر بدست می‌آید (شی و ابرهات، ۱۹۹۸):

$$v_{i,j}^{(t+1)} = \omega v_{i,j}^{(t)} + \rho_l \eta_l (x_{i,j}^{(best)} - x_{i,j}^{(t)}) + \rho_g \eta_g (x_{g,j}^{(best)} - x_{i,j}^{(t)}), \quad (\text{ب-۱۸})$$

که در آن $x_i^{(best)}$ موقعیت بهترین مکانی است که ذره مورد نظر در طول حرکت خود در آن قرار داشته و از رابطه زیر محاسبه می‌گردد:

$$x_i^{(best)} = \arg \min_x \left\{ \mathcal{J} \left(x_i^{(\tau)} \right) \right\}_{\tau=1}^t. \quad (\text{ب-۱۹})$$

همچنین $x_g^{(best)}$ نیز موقعیت بهترین مکانی است که کل اجتماع ذرات در طول حرکت خود در آن قرار داشته‌اند (بهترین سراسری ^{۳۲}):

$$x_g^{(best)} = \arg \min_x \left\{ \mathcal{J} \left(x_i^{(best)} \right) \right\}_{i=1}^N. \quad (\text{ب-۲۰})$$

به علاوه در رابطه (ب-۱۸) ω ضریب اینرسی ^{۳۳}، ρ_l ضریب یادگیری موضعی ^{۳۴} یا فردی ^{۳۵} و ρ_g ضریب یادگیری سراسری ^{۳۶} یا گروهی ^{۳۷} می‌باشند. همچنین η_l و η_g اعدادی تصادفی در بازه $[0, 1]$ با توزیع یکنواخت می‌باشند که متفاوت بودن تاثیر تجربیات بر عملکرد ذرات را پیاده‌سازی می‌کند. **کلرک و کندی (۲۰۰۲)** بر اساس یک سری محاسبات ریاضی مقادیر زیر را برای ضرایب اینرسی، یادگیری موضعی و سراسری پیشنهاد کردند:

$$\omega = 1, \rho_l = \rho_g = 1/4962. \quad (\text{ب-۲۱})$$

مرحله‌ی بعد از بدست آوردن سرعت ذرات جابجایی آن‌ها بر اساس سرعت بدست آمده و به روز رسانی

³²Global best

³³Inertia factor

³⁴Local learning factor

³⁵Individual learning factor

³⁶Global learning factor

³⁷Swarm learning factor

موقعیت‌ها می‌باشد. بر این اساس موقعیت جدید هر ذره به شکل زیر محاسبه می‌شود:

$$x_i^{(t+1)} = x_i^{(t)} + v_i^{(t+1)}. \quad (\text{ب-۲۲})$$

ب-۳ بهینه‌سازی چندهدفه

در بسیاری از مسائل بهینه‌سازی صنعتی، بیش از یک عامل در ارزیابی رفتار سیستم تاثیر دارد. از این رو بهینه‌سازی عموماً چند هدف را به طور همزمان دنبال می‌کند. مسائلی که در آن‌ها بیش از یک تابع هدف وجود دارد مسئله بهینه‌سازی چندهدفه^{۳۸} نام دارد. چالشی که در بهینه‌سازی مسائل چندهدفه وجود دارد نحوه ارزیابی بین طراحی‌های مختلف است چراکه معمولاً توابع هدف در تضاد با یکدیگر عمل کرده و امکان دستیابی به نقطه‌ای که تمام توابع را به صورت همزمان بهینه کند وجود ندارد. به عنوان مثال در بهینه‌سازی الگوریتم ژنتیک، تعریف عملگر محاسبه برازندگی در یک مسئله تک‌هدفه^{۳۹} به سادگی امکان‌پذیر خواهد بود حال آن که برای یک مسئله چندهدفه می‌تواند از پیچیدگی بالایی برخوردار باشد چرا که تعیین اولویت بین چند طراحی که هر کدام در برخی از توابع هدف عملکرد بهتری دارند به سادگی امکان‌پذیر نیست. علت این امر در واقع به فیزیک مسئله مرتبط می‌باشد. به عنوان مثال هدف از بهینه‌سازی خنک‌کاری پره توربین گاز، کاهش دمای بیشینه (و اندازه‌گرادیان دمای بیشینه) و دبی خنک‌کاری می‌باشد. حال سوالی که در اینجا وجود دارد این است که بین دو طراحی موجود که در یکی مقدار دبی خنک‌کاری و در دیگری مقدار دمای بیشینه مناسب‌تر است کدام یک به عنوان طراحی بهینه انتخاب شوند. برای حل این مشکل راه‌حل‌های گوناگونی پیشنهاد شده است. در ادامه ابتدا روش‌های سنتی به طور مختصر بررسی شده و سپس روش مرتب‌سازی غیر مغلوب^{۴۰} که در الگوریتم‌های تکاملی به کار می‌رود و در این رساله نیز مورد استفاده قرار گرفته شرح داده می‌شود.

ب-۳-۱ رویکردهای سنتی

رویکردهای سنتی عموماً بر پایه این ایده شکل گرفتند که مشکل ارزیابی‌های چندهدفه پیش از آن که در مرحله بهینه‌سازی مورد توجه قرار گیرد باید توسط طراح پاسخ داده شود. چرا که این طراح است که باید با

³⁸Multi-objective optimization

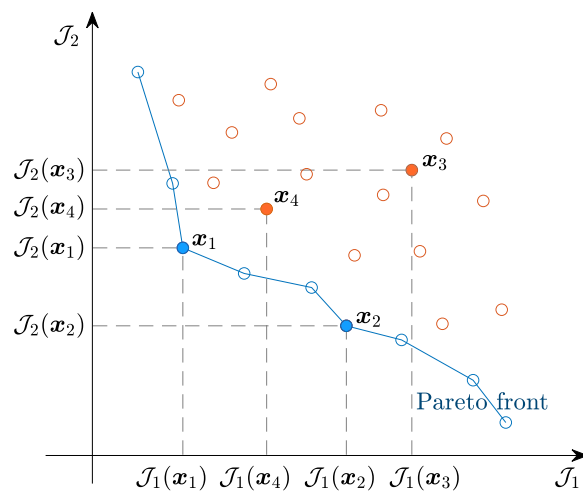
³⁹Single-objective optimization

⁴⁰Non-dominant sort

توجه به معیارهای موجود از توانایی انتخاب یک طرح از بین طرح‌های پیشنهادی برخوردار باشد و معیاری که توسط طراح امکان به کارگیری دارد در بهینه‌سازی نیز کم و بیش قابل پیاده‌سازی است. به عنوان نمونه یکی از راهکارهای موجود، در نظر گرفتن یک مشخصه به عنوان تابع هدف بهینه‌سازی و لحاظ نمودن سایر اهداف به صورت قیود بهینه‌سازی است. به عنوان نمونه در مسئله بهینه‌سازی خنک‌کاری پره توربین گاز می‌توان گفت مقدار دمای بیشینه به خودی خود اهمیتی ندارد و همین که دمای پره از مقدار مجاز تجاوز نکند کافی است. در اینصورت هدف از بهینه‌سازی کمینه‌سازی دبی خنک‌کاری با رعایت قید دمای بیشینه پره خواهد بود. روش دیگری که در مسائل بهینه‌سازی چندهدفه بسیار مورد توجه قرار می‌گیرد استفاده از ترکیب خطی توابع هدف به عنوان یک تابع هدف واحد است. در این رویکرد نیز ادعا می‌شود که طراح باید از ابتدای طراحی از میزان اهمیت نسبی هر مشخصه عملکردی آگاهی داشته باشد. بدین ترتیب می‌تواند با یک سری ضرایب وزنی میزان اهمیت هر مشخصه را در تابع هدف واحد تعیین کرده و بهینه‌سازی را با همان یک تابع انجام دهد.

روش‌های سنتی هرچند بسیار ساده بوده و از ویژگی سهولت در پیاده‌سازی برخوردارند با این حال ممکن است در عمل طراح را با محدودیت‌های جدی مواجه کند و طراحی‌های بهینه بدست آمده از دقت کافی و عملکرد مورد نیاز برخوردار نباشند. چرا که ممکن است شناخت رفتار دقیق سیستم پیش از ارائه طرح یا طرح‌های پیشنهادی عملاً غیر ممکن باشد. برای مثال در مسئله بهینه‌سازی خنک‌کاری توربین گاز، علت انتخاب دمای بیشینه پره به عنوان یک تابع هدف، تاثیر مستقیم آن در عمر پره می‌باشد. با این حال از آنجا که محاسبه عمر یک پره واقعی بسیار پیچیده است تخمین آن در هر مرحله بهینه‌سازی و لحاظ نمودن عمر به عنوان یک قید یا تابع هدف تقریباً غیر ممکن می‌باشد. از طرف دیگر رابطه عمر با دمای بیشینه مشخص نبوده و بسیاری از عوامل دیگر همچون مشخصات هندسی که خود متغیرهای طراحی هستند نیز روی آن تاثیر می‌گذارند. از این رو امکان تعریف دمای بیشینه مجاز پره در عمل وجود ندارد. راهکار ضریب وزنی نیز به دلایل مشابه سودمند نمی‌باشد. در چنین مواردی طراح ترجیح می‌دهد دست او در طراحی باز گذاشته شود و بجای یک طراحی بهینه، مجموعه‌ای از طراحی‌های بهینه به او پیشنهاد شود که با بررسی‌های دقیق‌تری یکی از آن‌ها را انتخاب کند. این ایده یکی از مبانی شکل‌گیری روشی است که در ادامه توضیح داده می‌شود.

ب-۳-۲ رویکرد مرتب‌سازی غیر مغلوب



شکل ب-۱: نمونه‌ای از مجموعه نقاط در فضای توابع هدف و انتخاب جبهه پارتو.

رویکرد مرتب‌سازی غیر مغلوب بر مبنای این ایده شکل گرفته است که در بهینه‌سازی چند هدفه ایجاد تمایز بین برخی طراحی‌ها امکان‌پذیر نیست و اساساً یک طراحی بهینه وجود ندارد. از این رو بجای یک نقطه بهینه یک مجموعه نقاط بهینه معرفی خواهد شد. این کار از طریق دسته‌بندی طراحی‌ها و رده‌بندی دسته‌ها صورت می‌گیرد به طوری که مجموعه طراحی‌های هر دسته نسبت به دسته‌ای که در مرتبه بعدی قرار گرفته مناسب‌تر بوده اما نقاط واقع در یک دسته تفاوتی از نظر ارزش عملکردی ندارند. در واقع همان‌طور که در بهینه‌سازی تک‌هدفه، یکی از روش‌های محاسبه برازندگی بر اساس تابع هدف رده‌بندی طراحی‌های پیشنهادی و اختصاص رتبه به هر طراحی بود، در این رویکرد نیز طراحی‌ها رده‌بندی می‌شوند با این تفاوت که دو یا چند طراحی ممکن است رتبه یکسانی داشته باشند. برای رسیدن به این منظور، **چانکونگ و هیمز (۱۹۸۳)** رویکرد مرتب‌سازی غیر مغلوب را برای مقایسه بین مجموعه‌ای از طراحی‌ها در مسائل چندهدفه که اصطلاحاً تصمیم‌گیری چندمعیاره ^{۴۱} نامیده می‌شود پیشنهاد دادند. اگر $\mathcal{T}(x)$ مدلی از رفتار سیستم باشد که در آن x مجموعه متغیرهای طراحی و $\mathcal{T}$ مجموعه مشخصه‌های عملکردی سیستم باشد. در اینصورت بنابر تعریف نقطه x_i بر x_j غلبه می‌کند $(x_i \prec x_j)$ اگر و فقط اگر تمامی مشخصات در نقطه x_i مطلوب‌تر از مقادیر متناظر در نقطه x_j باشد. به عنوان مثال در شکل ب-۱ که یک نمونه از مجموعه نقاط در فضای توابع هدف را نشان می‌دهد نقطه x_1 بر نقاط x_3 و x_4 غلبه می‌کند اما بر نقطه x_2 غلبه نمی‌کند. با این

⁴¹Multiple-criteria decision-making (MCDM)

تعریف، برای یک مجموعه طراحی پیشنهادی، نقطه یا نقاطی که هیچ طراحی دیگری بر آن‌ها غلبه نداشته باشد، نقاط غلبه نشده یا غیر مغلوب ^{۴۲} بوده و به مجموعه نقاطی که همگی غیر مغلوب باشند نقاط بهینه پارتو ^{۴۳}، جبهه بهینه پارتو ^{۴۴} یا جبهه اول پارتو گفته می‌شود (هنز، ۱۹۸۸؛ هورن و دیگران، ۱۹۹۴). رتبه این نقاط در رده‌بندی طراحی‌ها برابر رتبه اول می‌باشد. برای مثال در شکل ب-۱ نقاط آبی رنگ نشان‌دهنده جبهه پارتو می‌باشد. لازم به ذکر است هیچ نقطه‌ای بر نقاط پارتو غلبه نمی‌کند اما لزومی ندارد که نقاط واقع بر جبهه پارتو بر تمامی نقاط غلبه کنند. برای نمونه در شکل ب-۱ نقطه x_2 بر نقطه x_3 غلبه می‌کند اما بر نقطه x_4 غلبه نمی‌کند. بدین ترتیب برای محاسبه نقاط جبهه دوم کافی است نقاط جبهه بهینه پارتو از مجموعه طراحی‌های مورد نظر حذف شود (در اینصورت فقط نقاطی باقی می‌مانند که توسط جبهه اول پارتو مغلوب واقع شده‌اند) و نقاط غلبه نشده از بین نقاط باقی مانده محاسبه شوند. این نقاط جبهه دوم را تشکیل می‌دهند و در رده دوم رتبه‌بندی طراحی‌ها قرار دارند. این روند تا تشکیل تمامی جبهه‌های ممکن و رتبه‌بندی تمام نقاط ادامه خواهد داشت. سرینیواس و دب (۱۹۹۴)؛ دب و دیگران (۲۰۰۲) از این ایده در بهینه‌سازی الگوریتم ژنتیک استفاده کردند و نسخه اول و دوم الگوریتم ژنتیک مرتب‌سازی غیر مغلوب ^{۴۵} را توسعه دادند که در سایر الگوریتم‌های بهینه‌سازی تکاملی نیز قابل استفاده است. فرض کنید مسئله بهینه‌سازی تک‌هدفه (۳-۸۴) به صورت زیر تبدیل به یک مسئله چندهدفه گردد:

$$x^* = \arg \min_x \mathcal{J}(x), \quad \mathcal{N}(x^*) \leq 0, \quad (\text{ب-۲۳})$$

که در آن $\mathcal{J} = \{\mathcal{J}_1, \mathcal{J}_2, \dots, \mathcal{J}_Q\}$ مجموعه Q تابع هدف می‌باشد. حال تنها چالش در بهینه‌سازی این مسئله، محاسبه برازندگی یا تابع هدف معادل در هر نقطه می‌باشد. برای این منظور در هر مرحله از الگوریتم بهینه‌سازی مقدار توابع هدف برای تمامی اعضای جمعیت مورد نظر ($\mathcal{P}$) محاسبه شده و رتبه افراد با استفاده از رویکرد مرتب‌سازی غیر مغلوب بدست می‌آید. لازم به ذکر است در این مسئله برای هر $x_i, x_j \in \mathcal{P}$ می‌توان گفت $x_i \prec x_j$ اگر:

$$\mathcal{J}_s(x_i) \leq \mathcal{J}_s(x_j), \quad \forall s \in \{1, 2, \dots, Q\}. \quad (\text{ب-۲۴})$$

^{۴۲}Non-dominated^{۴۳}Pareto-optimal points^{۴۴}Pareto-optimal front^{۴۵}Non-dominated sorting genetic algorithm (NSGA)I,II

همچنین جبهه پارتو بهینه ($\mathcal{F}_1$) به صورت زیر تعریف می‌گردد:

$$\mathcal{F}_1 = \{x_i \in \mathcal{P} : x_i \prec x_j; \forall x_j \in \mathcal{P}\}. \quad (\text{ب-۲۵})$$

بدین ترتیب سایر جبهه‌ها نیز به طور مشابه بدست می‌آیند:

$$\mathcal{F}_k = \{x_i \in \mathcal{P} \setminus \mathcal{F}_{\{1,2,\dots,k-1\}} : x_i \prec x_j; \forall x_j \in \mathcal{P} \setminus \mathcal{F}_{\{1,2,\dots,k-1\}}\}, \quad (\text{ب-۲۶})$$

$$\mathcal{F}_{\{1,2,\dots,k-1\}} = \bigcup_{l=1}^{k-1} \mathcal{F}_l.$$

در نهایت برازندگی هر فرد به عنوان مثال برابر با تعداد جبهه‌های تشکیل شده منهای شماره جبهه هر فرد به علاوه یک فرض می‌شود. ادامه روند بهینه‌سازی نیز همچون الگوریتم تک‌هدفه خواهد بود.

ب-۳-۳ نقاط معیار

در بخش گذشته رویکرد مرتب‌سازی غیر مغلوب معرفی گردید و نشان داده شد که در این رویکرد بجای یک نقطه بهینه یک مجموعه نقاط بهینه تحت عنوان جبهه بهینه پارتو بدست می‌آید. با این حال در بسیاری از موارد وجود یک معیار جهت بررسی فرآیند بهینه‌سازی و جبهه بهینه پارتو ضروری به نظر می‌رسد. از جمله این موارد می‌توان به بررسی همگرایی بهینه‌سازی، مقایسه دو جبهه پارتو، انتخاب موقعیت $x_g^{(best)}$ در بهینه‌سازی اجتماع ذرات (رابطه ب-۲۰) و یا پیشنهاد یک نقطه بهینه به طراح اشاره کرد. این معیار می‌تواند یک یا چند نقطه از فضای طراحی به نمایندگی از جبهه پارتو باشد. از این رو **اِرگات و تفلدپودل** (۲۰۰۳) نقاط زیر را به عنوان معیاری از منحنی پارتو معرفی کردند:

۱. نقطه حسیض^{۴۶}:

نقطه‌ای (فرضی) در فضای توابع هدف ($\mathcal{J}^N$) است که مقادیر توابع هدف در آن نقطه برابر با بدترین مقادیر متناظر روی جبهه‌ی بهینه پارتو یا به عبارتی کران بالای مجموعه بردارهای تابع هدف جبهه پارتو می‌باشد:

$$\mathcal{J}^N = \{\mathcal{J}_1^N, \mathcal{J}_2^N, \dots, \mathcal{J}_Q^N\}, \mathcal{J}_i^N = \sup_{x \in \mathcal{F}_1} \{\mathcal{J}_i(x)\}. \quad (\text{ب-۲۷})$$

۲. نقطه ایده‌آل^{۴۷}:

⁴⁶Nadir point

⁴⁷Ideal point

نقطه ای در فضای توابع هدف ($\mathcal{J}^I$) که مولفه های آن در این فضا برابر با بهینه ترین مقادیر توابع هدف بر روی نقاط جبهه ی پارتو یا کران پایین مجموعه بردارهای توابع هدف جبهه پارتو می باشد:

$$\mathcal{J}^I = \{\mathcal{J}_1^I, \mathcal{J}_2^I, \dots, \mathcal{J}_Q^I\}, \mathcal{J}_i^I = \inf_{x \in \mathcal{F}_1} \{\mathcal{J}_i(x)\}. \quad (\text{ب-۲۸})$$

با توجه به تعریف بزرگ ترین کران بالا، نقطه ایده آل ممکن است بر تمامی نقاط جبهه پارتو غلبه داشته باشد یا نداشته باشد. از این رو بنا بر برخی دلایل عددی و در عمل، تعریف یک نقطه معیار که بر تمامی نقاط جبهه پارتو غلبه کند مفید است و در نتیجه نقطه آرمانی ^{۴۸} به عنوان نقطه ای در فضای توابع هدف نزدیک به نقطه ایده آل تعریف می شود که این ویژگی را داشته باشد:

$$\mathcal{J}^u = \{\mathcal{J}_1^u, \mathcal{J}_2^u, \dots, \mathcal{J}_Q^u\}, \mathcal{J}_i^u = \mathcal{J}_i^I - \epsilon_i, \epsilon_i > 0. \quad (\text{ب-۲۹})$$

۳. **نقطه بهینه توافقی** ^{۴۹}: نقطه بهینه توافقی که از آن تحت عنوان نقطه بهینه منتخب ^{۵۰} نیز یاد می شود به شکل های گوناگون در فضای متغیرهای طراحی تعریف می شود. یکی از رایج ترین تعاریف در این زمینه، نزدیک ترین نقطه در جبهه پارتو به نقطه ایده آل می باشد:

$$x^* = \arg \min_{x \in \mathcal{F}_1} \|\mathcal{J}(x) - \mathcal{J}^I\|, \quad (\text{ب-۳۰})$$

که در آن $\|\cdot\|$ عملگر نرم ^{۵۱} روی فضای $\mathbb{R}^Q$ بوده و به عنوان مثال می تواند نرم ℓ_q باشد:

$$\|z\|_q = \left(\sum_{i=1}^Q |z_i|^q \right)^{\frac{1}{q}}, \quad z = \{z_1, z_2, \dots, z_Q\}. \quad (\text{ب-۳۱})$$

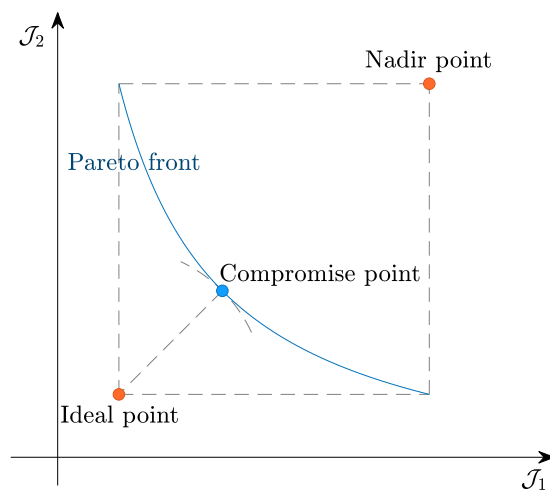
لازم به ذکر است تعاریف دیگری نیز برای نقطه بهینه توافقی پیشنهاد شده است. به عنوان مثال دورترین نقطه روی جبهه پارتو نسبت به نقطه حضيض و یا محل تلاقی جبهه پارتو با خط واصل بین نقطه ایده آل و نقطه حضيض دیگر تعاریف در این زمینه می باشند. در این رساله، برای تعریف نقطه بهینه توافقی از رابطه (ب-۳۰) در فضای بی بعد توابع هدف با استفاده از نرم ℓ_2 استفاده شده است. برای آشنایی با این نقاط، شکل ب-۲ یک منحنی پارتو نمونه را به همراه نقاط حضيض، ایده آل و بهینه توافقی نشان می دهد.

⁴⁸Utopian point

⁴⁹Compromise optimum point (COP)

⁵⁰Selected optimum point (SOP)

⁵¹Norm



شکل ب-۲: نمونه‌ای از نقاط معیار.