

۱-

الف) برای دستیابی به فرایند تکامل، دو ویژگی اساسی برای موجودات ضروری است: تنوع و انتخاب طبیعی. هر دوی این ویژگی‌ها به طور ترکیبی بر روی جمعیت موجودات تأثیر می‌گذارند. تنوع، به معنای وجود تنوع ژنتیکی در جمعیت موجودات است. این به این معناست که بین فرزندان تولید شده توسط والدین، تفاوت‌های ژنتیکی وجود داشته باشد. تنوع ژنتیکی امکان به وجود آمدن ترکیبات جدید و مختلفی از صفات و ویژگی‌ها را فراهم می‌کند. به عبارتی، تنوع ژنتیکی یک جامعه را مستعد‌تطور و سازگاری با محیط زیست می‌کند.

اما انتخاب طبیعی، به معنای فرایندی است که در آن موجودات با صفات و ویژگی‌های مفید و مناسب توانایی بقا و تکثیر بیشتری دارند. در طبیعت، موجوداتی که بهتر به محیط زیست خود سازگاری داشته و در مواجهه با چالش‌های زیستی موفقیت بیشتری داشته‌اند، احتمال بقا و انتقال ژن‌های خود را به نسل بعدی افزایش می‌دهند. در نتیجه، صفات و ویژگی‌های مفیدتر از طریق ارث برده می‌شوند و باعث ایجاد تغییرات تکاملی در جمعیت می‌شوند.

بنابراین، تنوع و انتخاب طبیعی در واقع همکاری می‌کنند و در تضاد نیستند. تنوع ژنتیکی در جمعیت به موجودات امکان می‌دهد تا با تغییرات محیطی و چالش‌های جدید، صفات و ویژگی‌های مناسب را تولید کنند. سپس انتخاب طبیعی به عنوان مکانیزمی طبیعی، موجودات با صفات مفید را از میان جمعیت برگزیند و این صفات را به نسل‌های آینده منتقل کند. در واقع، انتخاب طبیعی به عنوان عاملی فعال در تکامل و اصلاح جمعیت‌ها عمل می‌کند و تنوع را حفظ می‌کند.

از نظر مفهومی، می‌توان گفت تنوع متناظر با جست‌وجوی عمومی و اکتشاف است. زیرا تنوع، موجودات را قادر می‌سازد که به دنبال حلقه‌های جدیدی از آزمون و خطا بگردند و راه‌های جدیدی برای سازگاری با محیط زیست پیدا کنند. از سوی دیگر، انتخاب طبیعی متناظر با جست‌وجوی محلی و استخراج است. زیرا در این فرایند، موجودات با شایستگی بالاتر و قابلیت بقا و تکثیر بیشتر، به طور طبیعی انتخاب شده و ویژگی‌های مفید برای بقا در محیط زیست خاص خود استخراج می‌شوند.

در نهایت، می‌توان گفت که تنوع و انتخاب طبیعی همزمان عامل‌هایی هستند که به تکامل زیستی کمک می‌کنند. تنوع ژنتیکی از طریق اکتشاف و جست‌وجوی عمومی، تولید صفات جدید را امکان‌پذیر می‌کند، در حالی که انتخاب طبیعی از طریق استخراج و جست‌وجوی محلی، صفات مفید را از بین تنوع موجود برگزیده و در جمعیت تثبیت می‌کند. این دو ویژگی به همراه با سایر عوامل تکاملی، نظامی پویا را برای تنوع زنده‌ها در طبیعت فراهم می‌کنند.

ب) همگرایی زودرس و ساکن شدن، دو پدیده مختلف در تکامل موجودات هستند که می‌توانند تنوع را تحت تأثیر قرار دهند.

همگرایی زودرس به معنای تمایل جمعیت موجودات به تمرکز بر روی یک مجموعه محدود از صفات است. در نمونه ارائه شده، صفات با شایستگی 9 به طور چشمگیری بیشتر انتخاب می‌شوند و در نتیجه جمعیت به سمت موجودات با این صفت همگرا می‌شود. این موضوع می‌تواند منجر به کاهش تنوع

ژنتیکی در جمعیت شود، زیرا صفات دیگر با شایستگی‌های کمتر به طور کمتری انتخاب شده و در نتیجه در طول زمان کمتر به نسل‌های بعدی منتقل می‌شوند. با ادامه این پروسه، تنوع ژنتیکی کاهش می‌یابد و جمعیت به سمت همگرایی به یک مقدار محدود از صفات پیش می‌رود. ساکن شدن در واقع یک حالت است که در آن تنوع ژنتیکی در جمعیت حفظ می‌شود و تغییرات بسیار کمی در صفات رخ می‌دهد. در مثال دوم، شایستگی‌ها به طور نزدیکی به یکدیگر واقع شده‌اند و هیچ تفاوت چشمگیری بین آن‌ها وجود ندارد. در این حالت، هر کدام از موجودات با احتمال یکسانی انتخاب می‌شوند و جمعیت به صورت تقریباً ثابت باقی می‌ماند. این موضوع می‌تواند منجر به کاهش تغییرات تکاملی و کند شدن فرایند تکاملی در جمعیت شود. در نتیجه، همگرایی زودرس و ساکن شدن هر دو می‌توانند منجر به کاهش تنوع در جمعیت شوند. همگرایی زودرس منجر به همگرایی به یک مقدار محدود از صفات با شایستگی بالا می‌شود، در حالی که ساکن شدن به حفظ تنوع ژنتیکی و کاهش تغییرات در جمعیت منجر می‌شود. برای حفظ تنوع، مهم است که انواع مختلفی از صفات و شایستگی‌ها در جمعیت وجود داشته باشد و به اندازه کافی تنوع ژنتیکی حفظ شود.

-۲

(الف) در واقع یک روش تکاملی نیست، زیرا در این روش تنها یک نسل از فرزندان تولید می‌شود و هیچ فرزند دیگری تولید نمی‌شود. در واقع، در این روش تنها یک فرزند براساس والد فعلی تولید می‌شود و فرزندان گذشته نادیده گرفته می‌شوند. برای تکامل موثر، تنوع و اکتشاف لازم است. در این روش تنوع به شدت محدود است زیرا تنها یک فرزند براساس والد فعلی تولید می‌شود. این باعث می‌شود که جمعیت به سمت همگرایی زودرس حرکت کند و تنوع ژنتیکی در جمعیت کاهش یابد.

(ب) این روش در واقع یک روش تصادفی است و به تنهایی به تکامل موجودات منجر نمی‌شود. در این روش، تنوع وجود دارد زیرا تصادفی یک موجود انتخاب می‌شود ولی به دلیل حذف تنها یک موجود، انتخاب طبیعی در جمعیت وجود ندارد.

(ج) این روش یک الگوریتم تکاملی است که در آن μ نسل جدید از فرزندان تولید می‌شود و نسل قبلی به کاملیت جایگزین می‌شود. این الگوریتم به عنوان یک الگوریتم نسلی شناخته می‌شود. در این روش تکاملی، تنوع در جمعیت به دلیل تولید فرزندان جدید ایجاد می‌شود. با تولید μ فرزند در هر نسل، این فرزندان تنوع ژنتیکی جدیدی را به جمعیت اضافه می‌کنند. هر فرزند با ترکیب صفات و ویژگی‌های والدین خود، صفات جدید و متفاوتی را به جمعیت می‌آورد. این باعث حفظ و افزایش تنوع در جمعیت و ایجاد پتانسیل برای بهبود صفات شایسته‌تر می‌شود.

همچنین، در این الگوریتم، انتخاب طبیعی نیز وجود دارد. زیرا از بین μ فرزند تولید شده در هر نسل، μ فرزند با بهترین عملکرد یا شایستگی انتخاب می‌شوند تا وارد نسل بعدی شوند. این انتخاب بر اساس عملکرد و یا شایستگی‌های مشخصی که در ارزیابی هر فرزند استفاده می‌شود، صورت می‌گیرد. این باعث می‌شود تا فرزندان با شایستگی بالاتر در نسل‌های آینده حضور داشته باشند و تکامل به سمت بهبود صفات و عملکرد بهتر پیش برود.

د) این روش یک الگوریتم تکاملی است که در هر نسل μ والدین تولید می‌شود و یک فرزند جدید نیز ایجاد می‌شود. از بین مجموعه والدین و فرزند جدید، موجودیتی انتخاب می‌شود که عملکرد بدترین عضو نسل قبل را داشته باشد و در نسل جدید حذف می‌شود. این الگوریتم نیز به عنوان یک الگوریتم **steady state** شناخته می‌شود.

تولید یک فرزند جدید در هر نسل و حذف بدترین عضو نسل قبل، به جمعیت اجازه می‌دهد که تجربه و شایستگی را در طول زمان جمع دهد. با این روش، جمعیت بهبود یافته و به سمت شایستگی بالاتر پیشرو می‌کند. انتخاب طبیعی نیز در این الگوریتم وجود دارد، زیرا موجودیتی با عملکرد بدتر در نسل قبل، انتخاب و حذف می‌شود و موجودیتی با عملکرد بهتر و شایستگی بیشتر در جمعیت باقی می‌ماند. این الگوریتم نیز تغییرات آرامی و قدم به قدمی را به جمعیت می‌آورد. با تولید یک فرزند در هر نسل و حذف بدترین عضو، جمعیت به تدریج بهبود می‌یابد و به سمت شایستگی بالاتر پیشرفت می‌کند. این تغییرات آرامی و استمراری باعث می‌شود که جمعیت بهبود پیدا کند و به روند تکاملی پیشرو برود.

-۳-

روش **Roulette Wheel** (چرخش چرخان) یک روش انتخاب متناسب با شایستگی است که بر اساس احتمال و شاخص فیتنس موجودیت‌ها عمل می‌کند. برای اجرای این روش، ابتدا برای هر موجودیت، احتمالی به نام P_i محاسبه می‌شود که برابر با $(f_i / \sum f_i)$ است. در اینجا، $\sum f_i$ تعداد کل موجودیت‌ها و f_i شاخص فیتنس موجودیت مورد نظر است. سپس با استفاده از این احتمال‌ها، بازه‌هایی برای هر موجودیت ایجاد می‌شود که بازه‌ها متناسب با احتمال مربوط به هر موجودیت هستند. برای انتخاب موجودیت‌ها، ابتدا یک عدد تصادفی یکنواخت بین 0 و 1 تولید می‌کنیم. سپس با بررسی این عدد تصادفی در کدام بازه قرار می‌گیرد، موجودیت متناظر با آن بازه را انتخاب می‌کنیم. به عبارت دیگر، با توجه به بازه‌های ایجاد شده و عدد تصادفی تولید شده، موجودیتی که بازه‌ای که عدد تصادفی در آن قرار دارد، به آن تعلق دارد را انتخاب می‌کنیم.

$$F=[7, 1, 2, 2, 1, 6, 1, 8, 3, 4, 5]$$

حال برای هر موجودیت، احتمال P_i را محاسبه می‌کنیم:

$$P=[0.175, 0.025, 0.05, 0.05, 0.025, 0.15, 0.025, 0.2, 0.075, 0.1, 0.125]$$

سپس با تولید یک عدد تصادفی یکنواخت بین 0 و 1 (مانند 0.548)، بازه‌های متناسب با احتمال‌ها ایجاد می‌شود

حال با استفاده از اعداد تصادفی یکنواخت تولید شده (مانند 0.548)، موجودیت‌های متناظر با بازه‌هایی که در آن قرار دارند را انتخاب می‌کنیم:

Random_numbers = [0.548, 0.123, 0.789, 0.256, 0.634]

Selected = [8, 7, 4, 2, 8]

به این ترتیب، موجودیت‌های انتخاب شده با استفاده از روش Roulette Wheel به ترتیب زیر هستند:
[8, 7, 4, 2, 8]

روش SUS یک روش انتخاب متناسب با شایستگی است که بر اساس احتمال و شاخص فیتنس موجودیت‌ها عمل می‌کند. برای اجرای این روش، ابتدا برای هر موجودیت، احتمالی به نام P_i محاسبه می‌شود که برابر با $(f_i / (np))$ است. در اینجا، np تعداد کل موجودیت‌ها و f_i شاخص فیتنس موجودیت مورد نظر است. سپس با استفاده از این احتمال‌ها، خط کش دومی ایجاد می‌شود که طول آن برابر با $ns/1$ است (ns تعداد موجودیت‌هایی است که می‌خواهیم انتخاب کنیم).

سپس یک عدد تصادفی بین 0 و $ns/1$ انتخاب می‌کنیم و خط کش دوم را روی خط کش اول قرار می‌دهیم، از جایی که عدد تصادفی انتخاب شده شروع می‌کند. با این کار، خط کش دوم به طور مناسب بازه‌های انتخاب را روی خط کش اول تعیین می‌کند.

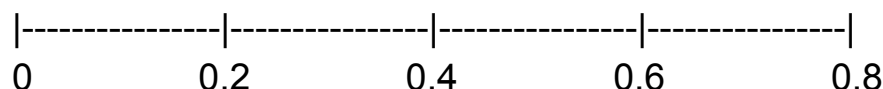
سپس با استفاده از اعداد تصادفی یکنواخت تولید شده (مانند 0.170، 0.370، 0.570، 0.770 و 0.970)، موجودیت‌های متناظر با بازه‌هایی که در آن‌ها قرار دارند، انتخاب می‌شوند.

$F=[7, 1, 2, 2, 1, 6, 1, 8, 3, 4, 5]$ $ns=5$ $np=11$

حال برای هر موجودیت، احتمال P_i را محاسبه می‌کنیم:

$P=[0.175, 0.025, 0.05, 0.05, 0.025, 0.15, 0.025, 0.2, 0.075, 0.1, 0.125]$

سپس خط کش دوم با طول $1/5$ ایجاد می‌شود:



حال با انتخاب اعداد تصادفی بین 0 و 0.2 (مانند 0.170)، انتخاب اول انجام می‌شود. سپس با انتخاب اعداد تصادفی بین 0.2 و 0.4 (مانند 0.370)، انتخاب دوم انجام می‌شود، و همین روند برای سه انتخاب بعدی تکرار می‌شود.

بنابراین، با اجرای روش SUS بر روی داده‌ها، موجودیت‌های انتخاب شده به ترتیب زیر هستند:

[7, 6, 8, 3, 5]

روش SUS نسبت به روش Roulette Wheel این مزیت را دارد که تضمین می‌کند که تعداد انتخاب‌ها بر اساس شاخص فیتنس صورت می‌گیرد. در واقع، $ns = N * pi$ را تضمین می‌کند (که در اینجا N تعداد کل انتخاب‌ها و ns تعداد انتخاب‌های انجام شده با احتمال pi است). در صورتی که تعداد انتخاب‌ها کم باشد، استفاده از روش SUS بهتر است زیرا تضمین می‌کند که انتخاب بر اساس شاخص فیتنس انجام شود.

۴-

الف) در تابع فیتنس که به صورت

$$\text{fitness}(\text{gen}) = \sum_{i=0}^{[n/2]-1} \neg(\text{gen}[i] \oplus \text{gen}[n-1-i])$$

تعریف شده است، رشته بیتی gen به طول n را می‌گیریم. در این تابع، هر بیت در موقعیت i را با بیت متقابلش در موقعیت $(n-1-i)$ مقایسه می‌کنیم. اگر بیت‌ها برابر بودند (0 و 0 یا 1 و 1)، مقدار $\neg(\text{gen}[i] \oplus \text{gen}[n-1-i])$ برابر با 1 خواهد بود و به fitness اضافه می‌شود. به عبارت دیگر، هرچه رشته بیت‌های متقارن بیشتری داشته باشد، تابع fitness مقدار بیشتری خواهد داشت. این تابع fitness به عنوان یک معیار برای ارزیابی کیفیت رشته بیتی gen استفاده می‌شود. بنابراین، هدف ما در فرآیند بهینه‌سازی می‌تواند با استفاده از الگوریتم‌های مختلفی مانند الگوریتم ژنتیک، جستجوی بهینه‌ی رشته بیتی با بیشترین مقدار تابع fitness باشد.

ب) در فرآیند بهینه‌سازی با استفاده از الگوریتم ژنتیک، ابتدا شایستگی (fitness) هر رشته از جمعیت اولیه محاسبه می‌شود. در اینجا سه رشته اولیه به شکل زیر و شایستگی متناظر آن‌ها آورده شده است:

رشته 1: 110001، شایستگی: 2

رشته 2: 011101، شایستگی: 1

رشته 3: 011000، شایستگی: 1

سپس، در مرحله انتخاب والدین با استفاده از روش Roulette Wheel Selection، احتمال انتخاب هر رشته محاسبه می‌شود. در اینجا، احتمال انتخاب برای هر رشته به شرح زیر است:

$$P(110001) = 0.5$$

$$P(011101) = 0.25$$

$$P(011000) = 0.25$$

پس از محاسبه احتمالات، سه بار عدد تصادفی بین 0 و 1 انتخاب می‌شود تا والدین انتخاب شوند. به عنوان مثال، فرض کنید در اینجا والدین زیر انتخاب شده‌اند:

والدین 1: 110001

والدین 2: 011101

والدین 3: 011000

در مرحله بازترکیبی، باید سه فرزند تولید کنیم. در اینجا، فرض می‌کنیم هر دو والد دو فرزند تولید کنند. بنابراین، سه بار جفت والد تصادفی انتخاب می‌شود و سپس با احتمال بازترکیبی (pc) که در اینجا برابر با 0.8 است، بازترکیبی روی جفت والد صورت می‌گیرد. نتایج بازترکیبی در اینجا به شرح زیر است:

والدین 1 و 2: 011001، 110101

والدین 2 و 3: 011101، 011000

والدین 1 و 3: 011001، 110000

سپس، با احتمال 0.25 هر بیت از فرزندان جهش می‌کند. در اینجا، فرض کنید فرزندان به شکل زیر جهش می‌کنند:

فرزند 1: 111101

فرزند 2: 011000

فرزند 3: 001100

فرزند 4: 011101

فرزند 5: 110100

فرزند 6: 001101

در انتها، با استفاده از روش **Roulette Wheel Selection**، سه فرزند از بین فرزندان انتخاب می‌شوند و به نسل بعدی منتقل می‌شوند. در اینجا، به عنوان مثال، سه فرزند زیر انتخاب می‌شوند و جایگزین نسل بعدی می‌شوند.

فرزند 1: 001100

فرزند 2: 111101

فرزند 3: 001101

با ادامه این فرآیند بهینه‌سازی با الگوریتم ژنتیک، نسل‌های جدید با شایستگی بالاتر تولید می‌شوند و تا رسیدن به راه حل بهینه ادامه می‌یابد.

-۵-

الگوریتم کلونی زنبورها یک الگوریتم بهینه‌سازی مبتنی بر تکامل است که بر اساس رفتار و ارتباط زنبورها در طبیعت توسعه یافته است. این الگوریتم برای حل مسائل بهینه‌سازی، به خصوص در مسائل مسیریابی، برنامه‌ریزی و بهینه‌سازی ترکیبیاتی مورد استفاده قرار می‌گیرد.

مراحل الگوریتم کلونی زنبورها به صورت زیر است:

1. مرحله مسیریابی:

- هر زنبور به صورت تصادفی یک گل را برای جمع آوری انتخاب می‌کند و از موقعیت اولیه خود شروع به حرکت می‌کند.

- زنبورها بر اساس قوانین خاصیت‌های بویایی و شایستگی، مسیر خود را بین گل و کندوی خود طی می‌کنند. در طول این مسیر، زنبورها اطلاعات محلی را در مورد شایستگی گل‌ها جمع‌آوری می‌کنند.

2. محاسبه شایستگی:

- هر زنبور پس از انجام مسیریابی، شایستگی مسیر خود را محاسبه می‌کند. این شایستگی می‌تواند بر اساس فاصله طی شده توسط زنبور، کمیتی مانند میزان گل‌های جمع‌آوری شده یا هر شاخص دیگری باشد.

3. اطلاعات‌رسانی:

- زنبورها اطلاعات خود را در مورد مسیر و شایستگی جمع‌آوری شده را به اشتراک می‌گذارند. این اطلاعات شامل موقعیت گل‌هایی است که زنبورها جمع‌آوری کرده‌اند و شایستگی آنها.

4. به‌روزرسانی مسیرها:

- با استفاده از اطلاعات جدید، زنبورها مسیرهای خود را بهبود می‌بخشند.

ممکن است زنبورها مسیرهای جدیدی را بر اساس اطلاعات به دست آمده انتخاب کنند یا مسیرهای قبلی خود را تغییر دهند.

5. تکرار مراحل 2 تا 4 تا رسیدن به شرایط خاتمه:

- الگوریتم به صورت تکراری مراحل 2 تا 4 را تکرار می‌کند تا شرایط خاتمه تعیین شده را برآورده کند. این شرایط می‌توانند مشخصاتی مانند تعداد تکرارها، تغییرات کمی در شایستگی یا تحقق یک شرط معین باشد.

در هر تکرار، زنبورها بهبودهایی در مسیرهای خود ایجاد می‌کنند و با تبادل اطلاعات، دانش محلی را با یکدیگر به اشتراک می‌گذارند. این ارتباطات و بهبودها به تدریج منجر به پیدایش یک مسیر بهینه برای زنبورها می‌شود که کمترین مسافت را طی می‌کند و به عبارتی عملکرد بهتری نسبت به روش‌های دیگر دارد.

مهمترین ویژگی الگوریتم کلونی زنبورها، قدرت تعامل و همکاری زنبورها با یکدیگر است. زنبورها بهبودهایی که در مسیرهای خود ایجاد می‌کنند را با هم به اشتراک می‌گذارند و از تجربیات همدیگر بهره می‌برند. این ویژگی باعث می‌شود که الگوریتم کلونی زنبورها به طور کلی به روش‌های موازی و توزیع شده منطبق باشد و قابلیت اعمال بر روی مسائل بزرگ و پیچیده را داشته باشد.

در استراتژی تکاملی $EA(1 + \lambda)$ ، هر دور یک والد انتخاب می‌شود و از آن والد یک فرزند ساخته می‌شود. سپس از بین مجموعه $1 + \lambda$ موجودیت (شامل والد و فرزندان)، بر اساس شایستگی، یک موجود به نسل بعدی منتقل می‌شود. برای توضیح روش $EA(1 + \lambda)$ برای مسئله مسیریابی کمترین مسیر (TSP)، می‌توانیم اقدامات زیر را انجام دهیم:

1. بازنمایی مسئله:

- هر پاسخ به مسئله مسیریابی TSP در واقع یک ترتیبی از شهرها است.
- برای مثال، فرض کنید n شهر داریم و شهر 1 را به عنوان شهر شروع انتخاب کرده‌ایم. یک پاسخ می‌تواند به صورت زیر باشد:

1 -> 5 -> 4 -> 6 -> 3 -> 7 -> 8 -> 2 -> 1

- در این مثال، از شهر 5 هم عبور کرده و در نهایت به شهر 1 بازگشته‌ایم.

2. تابع شایستگی:

- تابع شایستگی باید بر اساس طول مسیر متناظر با یک پاسخ (ترتیب شهرها) تعریف شود.
- برای محاسبه طول مسیر، می‌توان از ماتریس دو بعدی $dist$ استفاده کرد که فاصله بین هر دو شهر را نگه می‌دارد.
- تابع شایستگی می‌تواند به صورت زیر تعریف شود:

$$fitness(gen) = 1 / path(gen)$$

- که در آن $path(gen)$ مجموع فواصل بین تمام جفت شهرها در پاسخ gen است.
- با این تعریف، هر چه مسیر کمتری طی شود، شایستگی بیشتری دارد.

3. جمعیت اولیه:

- برای جمعیت اولیه، k پاسخ انتخاب می‌شوند که همه آنها با شهر اول (شهر شروع) آغاز می‌شوند.
- جمعیت اولیه به صورت تصادفی انتخاب می‌شود.

4. انتخاب والدین:

- بر اساس استراتژی $EA(1 + \lambda)$ ، در هر دور، یک والد انتخاب می‌شود.
- ابتدا، برای همه پاسخ‌های جمعیت اولیه، تابع شایستگی محاسبه می‌شود.
- سپس می‌توان از روش (roulette wheel) استفاده کرد تا با احتمال بیشتری پاسخ‌های با شایستگی بالاتر انتخاب شوند.

5. بازترکیبی:

- در این روش $EA(1 + \lambda)$ ، تنها یک والد وجود دارد و فرزند با جایگشت و تغییر ترتیب

ویژگی‌های والد (mutation) تولید می‌شود.

- یک روش جهت بازترکیبی می‌تواند شامل انتخاب دو بازه جدا از هم در یک پاسخ باشد و جایگشت این دو بازه را انجام دهد. در این حالت، بازه‌ها نباید شامل شهر اول (شهر شروع) باشند و طول بازه‌ها حداقل باید یک شهر باشد.

6. جهش:

- برای جهش، می‌توان از روش‌های مختلفی مانند swap دو شهر با یکدیگر استفاده کرد.

- برای مثال، می‌توان دو شهر متوالی را جابجا کنیم. این عملیات می‌تواند برای هر فرزند تولید شده تعدادی بار تکرار شود.

7. شرط خاتمه:

- شرایط خاتمه می‌تواند مختلف باشد، مانند تعداد مشخصی از ارزیابی‌ها، دستیابی به یک آستانه شایستگی خاص، همگرایی یا عدم تنوع.

- برای مسئله TSP، می‌توان شرایطی مانند رسیدن به یک مقدار خاص از شایستگی یا برآورده شدن یک آستانه شایستگی مشخص را در نظر گرفت.

به طور کلی، الگوریتم استراتژی تکاملی $EA(1 + \lambda)$ برای حل مسئله TSP می‌تواند به صورت زیر خلاصه شود:

1. تعیین تعداد جمعیت اولیه و ایجاد جمعیت اولیه تصادفی با شروع از یک شهر مشخص.

2. محاسبه تابع شایستگی برای هر پاسخ در جمعیت اولیه.

3. تکرار مراحل انتخاب والدین، بازترکیبی و جهش برای تولید فرزندان.

4. انتخاب والد بر اساس شایستگی و جایگزینی والد با فرزند در جمعیت اولیه.

5. بررسی شرط خاتمه و پایان الگوریتم در صورت برآورده شدن شرط مورد نظر.

6. بازگشت به مرحله 2 تا 5 تا زمانی که شرط خاتمه برآورده نشود.

توصیه می‌شود که در این الگوریتم، پارامترهای مختلفی مانند اندازه جمعیت اولیه، تعداد ژن‌ها، احتمال باز ترکیبی و احتمال جهش به صورت تجربی تنظیم شوند تا بهترین نتیجه برای مسئله TSP به دست آید.