

- **BBO** یک الگوریتم جست‌وجوی سراسری مبتنی بر جمعیت است که می‌تواند نواحی بهینه در فضای جست‌وجو را کشف کند.
- **RL** یک روش یادگیری مبتنی بر تعامل با محیط است که می‌تواند تصمیم‌گیری‌های مرحله‌به‌مرحله بهینه را یاد بگیرد.

✓ هدف از ادغام:

ترکیب قدرت اکتشاف **BBO** با قدرت یادگیری و بهینه‌سازی محلی **RL** برای بهبود کیفیت راه‌حل و سرعت همگرایی در مسئله زمان‌بندی.

ساختار کلی ادغام BBO و RL برای زمان‌بندی:

حالت ۱: استفاده از RL در درون BBO

در این حالت:

- هر کروموزوم (راه‌حل) در BBO با استفاده از یک عامل RL اصلاح (refine) می‌شود.
- BBO جمعیت راه‌حل‌ها را مدیریت می‌کند و RL هر راه‌حل را بهبود می‌دهد.

حالت ۲: استفاده از BBO برای راهنمایی RL

- BBO به عنوان تولیدکننده سیاست اولیه (initial policy) یا **exploration guide** برای عامل RL استفاده می‌شود.
- RL در طول زمان سیاست خود را با استفاده از داده‌های جمع‌آوری شده از BBO بهبود می‌دهد.

مراحل پیشنهادی برای پیاده‌سازی ادغام BBO + RL

۱. تعریف مسئله زمان‌بندی

برای مثال:

- زمان‌بندی کلاس‌ها، وظایف یا کارها.
- تعریف **state** و **reward, action** برای محیط RL

۲. طراحی محیط RL

- تعریف **states** (مثل جدول زمان‌بندی فعلی)، **actions** (اختصاص زمان یا مکان جدید) و **rewards** (مثلاً کاهش تداخل کلاس‌ها، استفاده بهتر از منابع).

۳. طراحی BBO

- هر جزیره (**habitat**) یک راه‌حل زمان‌بندی است.
- هر راه‌حل شامل توزیع کلاس‌ها / وظایف در زمان و مکان.

- استفاده از مهاجرت (migration) برای تبادل ویژگی‌ها میان راه‌حل‌ها.

۴. بهبود هر راه‌حل توسط RL

در هر نسل BBO :

- هر راه‌حل توسط عامل RL بهبود می‌یابد.
- عامل RL چند حرکت (actions) روی راه‌حل اجرا می‌کند تا کیفیت آن افزایش یابد.

۵. آموزش عامل RL

- استفاده از الگوریتم‌هایی مانند DQN ، PPO ، A2C یا Q-learning
- داده‌های تولیدشده از جمعیت BBO (که درون buffer ذخیره می‌شود) می‌تواند تجربه RL را بیشتر کند

چرخه ترکیبی:

برای هر نسل BBO:

۱. ارزیابی هر راه‌حل (fitness)
۲. بهبود هر راه‌حل با استفاده از عامل RL (با حرکت دادن در فضای actions)
۳. اعمال مهاجرت بین راه‌حل‌ها
۴. انتخاب بهترین راه‌حل‌ها
۵. آپدیت تجربه RL با استفاده از نتایج به دست آمده (عامل RL چند حرکت (actions) روی راه‌حل اجرا می‌کند تا کیفیت آن افزایش یابد).
۶. آموزش مجدد RL با استفاده از buffer