**n-vazir**

**import numpy as np**

**import random as rn**

**def creat\_ch(n):**

**ch = [rn.randint(1, n) for i in range(n)]**

**return ch**

**def fitness(ch):**

**n = len(ch)**

**f = 0**

**for i in range(n):**

**for j in range(i+1, n):**

**if ch[i] == ch[j] or abs(i-j) == abs(ch[i]-ch[j]):**

**f += 1**

**ch.insert(0, f)**

**return ch**

**def cross\_over(p1, p2):**

**n = len(p1)**

**r = rn.randint(1, n-1)**

**ch1 = p1.copy()**

**ch2 = p2.copy()**

**ch1[r:] = p2[r:]**

**ch2[r:] = p1[r:]**

**return ch1, ch2**

**def mution(ch):**

**n = len(ch)**

**r = rn.randint(0, n-1)**

**ch[r] = rn.randint(1, n)**

**return ch**

**def creat\_pop(npop, n):**

**pop = [creat\_ch(n) for i in range(npop)]**

**return pop**

**n=8**

**npop=10**

**pop=creat\_pop(npop,n)**

**pop**

**for i in range(npop):**

**pop[i]=fitness(pop[i])**

**pop.sort()**

**best = pop[0]**

**iter = 0**

**while best[0] != 0:**

**print(iter)**

**iter += 1**

**newpop = []**

**for i in range(0, npop, 2):**

**ch1, ch2 = cross\_over(pop[i][1:], pop[i+1][1:])**

**ch1 = mution(ch1)**

**ch2 = mution(ch2)**

**ch1 = fitness(ch1)**

**ch2 = fitness(ch2)**

**newpop.append(ch1)**

**newpop.append(ch2)**

**pop += newpop**

**pop.sort()**

**pop = pop[:npop]**

**best = pop[0]**

**best**

**ربات فوتبالیست**

**ربات فوتبالیست یک نوع ربات هوشمند است که با استفاده از فناوری هوش مصنوعی و تکنولوژی پیشرفته، طراحی شده است تا در زمین فوتبال یا محیط های مرتبط با این ورزش عمل کند. این ربات‌ها با استفاده از الگوریتم‌ها، شبکه‌های عصبی، و سایر فناوری‌های مرتبط با هوش مصنوعی، قابلیت تحلیل و پردازش اطلاعات مرتبط با بازی فوتبال را دارند.**

**ویژگی‌های مهم ربات فوتبالیست:**

**هوش مصنوعی (AI): از الگوریتم‌ها و مدل‌های هوش مصنوعی برای اتخاذ تصمیمات هوشمندانه در مواقع مختلف بازی استفاده می‌کنند.**

**تشخیص محیط: قابلیت تشخیص اجسام و شناخت محیط بازی، اعمال تاکتیک‌ها و حرکات مناسب را ممکن می‌سازد.**

**حرکت و عملکرد فیزیکی: توانایی انجام حرکات فیزیکی متناسب با اهداف تاکتیکی و موقعیت بازی.**

**آموزش و یادگیری: قابلیت آموزش و به‌روزرسانی مهارت‌ها و تاکتیک‌ها بر اساس تجربیات گذشته و شرایط جدید بازی.**

**تعامل با انسان: توانایی همکاری و تعامل با بازیکنان انسانی در تیم‌ها یا محیط‌ های تربیتی.**

**کاربردهای گسترده: علاوه بر تیم‌های حرفه‌ای، در تمرینات، بازی‌های آموزشی، و حتی برگزاری رویدادهای تفریحی و تحقیقاتی نیز کاربرد دارند.**

**از طراحی ربات فوتبالیست به منظور بهبود توانمندی و تجربه بازیکنان انسانی گرفته تا افزایش سطح آموزش و تفریح در حوزه فوتبال، این نوع ربات‌ها به دنیای ورزش فوتبال ارزش افزوده‌ای بسیار داده‌اند.**

**قوانین مسابقات ربات‌های فوتبالیست برای اطمینان از رقابت منصفانه و هماهنگی در این دوره‌های تحول‌آفرین تعیین شده‌اند. از میان قوانین اصلی این مسابقات می‌توان به موارد زیر اشاره کرد:**

**1. تعداد بازیکنان و اجزاء تیم: هر تیم باید تعداد مشخصی از ربات‌های فوتبالیست را در مسابقات حضور دهد و این تعداد ممکن است بر اساس قوانین هر مسابقه تعیین شود.**

**2. هوش مصنوعی و تصمیم‌گیری: ربات‌ها باید توانایی اتخاذ تصمیمات هوشمندانه در میدان بازی را داشته باشند و به تعلیمات تیم تطابق دهند.**

**3. تشخیص محیط: قابلیت تشخیص اجسام و محیط بازی برای اجرای حرکات و تاکتیک‌های مناسب.**

**4. فیزیک شبیه‌سازی: در برخی موارد، مسابقات ممکن است از شبیه‌سازی‌های فیزیکی استفاده کنند تا شرایط واقعی بازی فوتبال را تا حد امکان بازسازی کنند.**

**5. موارد ایمنی: اطمینان از ایمنی ربات‌ها و افراد حاضر در محیط مسابقه.**

**6. ارتباط و تعامل: توانایی ربات‌ها در ارتباط با هم و با تیم انسانی، اطلاع‌رسانی و هماهنگی مناسب در طول بازی.**

**7. اخلاقیات مسابقه: تعیین اصول اخلاقی برای تضمین رفتار مناسب تیم‌ها و حفظ انضباط در مسابقات.**

**همه این قوانین به منظور ایجاد یک فضای رقابتی فرهنگی و فناورانه برای توسعه هوش مصنوعی و رباتیک در حوزه ورزش فوتبال ایجاد شده‌اند.**

**CSP**

**همانطور که میدانید مسئله ها میتوانند از طریق جست وجو در فضایی از حالتها حل شوند. این حالتها را میتوان با ابتکارهای خاص دامنه ارزیابی، و تست کرد که آیا آنها حالتهای هدف هستند یا خیر. اما از نظر الگوریتم جست وجو هر حالت یک جعبه سیاه است که ساختار داخلی آن قابل تمیز دادن نیست.**

**در اینجا روشی داریم که تعداد زیادی از مسئله ها را با کارایی بیشتری حل میکند. برای هر حالت، از مجموعه ای از متغیرها که هر کدام دارای یک مقدار است، استفاده میکنیم. مسئله وقتی حل میشود که هر متغیر دارای مقداری باشد که تمام محدودیت های روی آن متغیر را برآورده کند (آن محدودیت را ارضا کند). مسئله ای که به این روش توصیف میشود، مسئله ارضای محدودیت یا CSP نام دارد.**

**الگوریتم جست وجوی CSP از ساختار حالت استفاده میکند و به جای ابتکارهای خاص مسئله، از ابتکارهای همه منطوره برای حل مسئله های پیچیده بهره میگیرند. ایده ی اصلی این است که با مشخص کردن ترکیب هایی از متغیر/ مقدار که محدودیت ها را نقض میکنند، بخش های بزرگی از فضای جست وجو بطور همزمان حذف شوند.**