**Introduction**

进化算法（Evolutionary Algorithms，EAs）是解决强化学习（Reinforcement Learning，RL）和基准测试（Benchmarking）等领域中非常重要的一种算法。在这些领域中，优化搜索的目标往往涉及到探索与开发之间的平衡，以及对大规模搜索空间的高效搜索。进化算法以其种群化的特性和对搜索空间的全面探索而闻名，逐渐成为解决这些复杂问题的首选方法之一。

Negatively Correlated Natural Evolution Strategy (NCNES)是论文[1]中提出的一种基于 Negatively correlated search (NCS)的进化算法(EA)，使用迭代的方式探索得出最优解。该算法的驱动思想是创建一些具有一定多样性的种群可以更有利于求得最优解。NCNES会测算不同种群之间的距离来保证种群具有一定的多样性，在综合了已有种群的评估价值和多样性分布后，NCNES会定向产生下一代种群的概率分布值，并据此生成下一代的种群。基于此，NCNES的特点是重点关注相邻两代之间的解的概率分布在解空间中的多样性，并有意识地维持发散性地全局搜素。

具体来说，Negatively Correlated Natural Evolution Strategy (NCNES)将种群明确划分为多个子种群。每个子种群的演化被视为一个独立的搜索过程，并由传统的进化算法进行开发利用。同时，通过驱使它们的概率分布呈负相关，这些搜索过程协调地探索不同的搜索空间。NCNES对于NCS的显著提升是，相比前者仅仅只是提出了模糊的负相关探索的概念，后者对其中的数学过程进行了详细地建模，从初始化，划分种群开始，直到每一轮迭代最后的梯度更新过程，NCNES都制定了严谨的数学公式模型，并根据模型训练进度制定了不同的学习率公式，从各方面都完善了负相关搜索思想的策略，使得它成为一个严谨的算法。

在优化方面，NCNES通过同时对两个模型进行梯度下降，得到的结果是负相关自然进化策略，它能够形成并行探索搜索行为，不同的搜索过程将并行地演化到不同但有前景的搜索空间区域。在项目后期，会引入论文[2]中的Cooperative Coevolution，CC策略，通过将决策变量划分为多个独立的组并分别处理较小的子问题（即决策变量组）来扩大EA的规模。

为了验证优化后的NCNES的表现，我将使用CEC 2017 Benchmark对其进行测试验证，CEC2017测试函数集的使用相当广泛，它对算法有一定挑战性且学术上认可度较高。测试集所有函数都进行了旋转和位移，增加了算法寻优难度。函数主要包含30个测试函数：其中F1、F3为没有局部最小只有全局最小的单峰函数，这些单峰函数能检验算法的收敛能力；F4-F10为具有局部极值点的多峰函数，这些多峰函数用于测试算法跳出局部最优的能力；F11-F20是经旋转/位移后包含三个或以上的CEC2017基准函数的混合函数，每个子函数被赋予一定权重；F21-F30是由至少三个混合函数或CEC 2017基准函数旋转移位后组成的复合函数，每个子函数有额外偏置值和一个权重，这些组合函数进一步增加了算法的优化难度。

本文的主要内容将会是对于NCNES算法的详细描述，并介绍对其改进的地方，然后会介绍引入的 Cooperative Coevolution (CC) 框架，最后通过列出算法在CEC 2017 Benchmark等基准测试上的结果分析其优势和需要改进的不足之处。

**NCNES**

对于负相关搜索（NCS）的由来，即如何重新思考种群在搜索中的作用。在组队合作的过程中，人们普遍会认为种群之间需要进行有效的信息共享，来避免重复劳动，以提升信息搜索的效率和能力，但一个尚未解决的问题是要分享什么信息以及如何分享。负相关的思想通过模仿人类的合作方式，要求种群中的个体具有不同的搜索行为，以避免重复搜索搜索空间的同一区域。每种搜索行为定义了后代如何基于其父代进行采样，所以我使用概率分布来表示每一代在概率分布中的位置。利用分布之间的数学相关性来统计建模种群之间的多样性。因此，通过明确地驱使多个概率分布呈负相关，本算法可以有效地控制下一个种群的多样性。

**数学模型**

NCNES的基本思想是将所有的解分为lamda个子种群，每个子种群内部使用同一个简单高斯分布函数采样。迭代地过程则可以描述为以适应性评估值和多样性评估值为目标，通过传统的进化算法进行独立地演化。因此，显然地，新的分布应该能够采样出具有高适应度值的新解。此外，新的分布应该与现有分布之间有更少的重叠（相关性），以便它们可以用来采样解空间的不同区域。

接下来，我们开始介绍进化的具体过程。正如前文所说，我们的solutions由指定的概率分布函数产生。在此，统一起见，我们设定所有分布为 Gaussian distribution, while the parameters of the distribution, e.g., mean and covariance, can be different. 因此，在初始状态，在所有种群都使用同一种概率密度函数的情况下，我们可以有以下函数

我们只需最大化这个单一的适应性函数。但是，由于算法需要同时对解空间的多个区域进行并行探索，所以以下我们引入第二个高斯分布函数，需要注意的是，NCNES强调不同搜索方向的负相关性，所以函数应该是：

其中，xxxxx代表第i个分布和第j个分布的相关性值, 样本分布相距越近，相关性值越大

现在我们将分布情况推广到多个种群，多种分布，我们将会得到：

现在我们初步地获得了优化的目标，它由两个加数组成，我们称其为F和D。形象来说，F的作用是根据当前的分布函数，评估给出当前的种群的解的表现值的期望；D的作用是构建一个当前本种群与其它种群的分布模型。也就是说，对于每一个种群，在计算D值的时候需要构建出它距离其它所有lamda-1个种群的距离，并尽可能将其最大化，由此，D可以表示为：

现在，由于我们已经把待优化的因素都并入了唯一的目标J，我们唯一的目标就是最大化J的值以获得演化问题的最优解。又因为，我们所定义的多种群的并行搜索过程在优化过程中是独立的，所以我们可以对 J 进行对每个 θ\_i 的偏导数下降

最后有一点值得一提，根据之前列出的公式，在处理不同的问题时，对应的fitness value会相差很大，所以我们很有必要采取措施来统一这个值的范围。我们对此的方式是，首先求出这个种群中所有\mu个解的fitness value，然后根据fitness value从小到大给他们排名，π(k) 表示第 k 个解的排名，然后根据以下公式对每个解的fitness value重新赋值

另外，对于学习率的更新，就是如何调整步长参数ηm和ηΣ。具体来说，这里采用了一种策略来动态地调整这两个参数，使其随着搜索的进行而逐渐减小。公式中的参数Tmax代表了整个搜索过程的总时间预算，而Tcur表示当前已经消耗的时间。e是自然常数。ηinitm和ηinitΣ分别是步长参数的初始值。公式通过一个指数衰减函数，将这两个步长参数从初始值逐渐减小到零。

同时，我也加入了$\phi$值的衰减，我希望模型在搜索前期有一定地关注多样性指标，而在模型训练后期才去重点关注fitness value。这样更符合模型搜索，学习的常理

总结一下，NCNES是基于多元高斯分布的进化算法；每个子种群由一个分布来驱动其迭代优化；通过所提出的多元高斯分布负相关的多样性模型，多个高斯分布之间形成负相关。因此，将其命名为“Negatively Correlated Natural Evolution Strategies”（NCNES）。以下伪代码中列出了对于NCNES的详细步骤。

实验

Benchmark2017

在测试环节，我们将总评估次数设置为$10000\*D$，每次从初始化开始训练迭代所有的种群，直到该种群中所有的解的评估次数之和为$10000\*D$为一次搜索过程，对于每一个test benchmark function, 我们进行十次相同的搜索过程，并对结果求mean均值和协方差，以此反应搜索算法的表现

CEC2017测试函数集的使用相当广泛，它对算法有一定挑战性且学术上认可度较高。测试集所有函数都进行了旋转和位移，增加了算法寻优难度。函数主要包含29个测试函数：

其中F1、F3为没有局部最小只有全局最小的单峰函数，这些单峰函数能检验算法的收敛能力；F4-F10为具有局部极值点的多峰函数，这些多峰函数用于测试算法跳出局部最优的能力；F11-F20是经旋转/位移后包含三个或以上的CEC2017基准函数的混合函数，每个子函数被赋予一定权重；F21-F30是由至少三个混合函数或CEC 2017基准函数旋转移位后组成的复合函数，每个子函数有额外偏置值和一个权重，这些组合函数进一步增加了算法的优化难度。（其中原函数集的F2函数被官方删除了，原因是F2不稳定）

CEC2022是评价算法性能和验证其求解复杂OP能力的有效方法。它由12个基准函数组成，包括单峰、多峰、混合和复合函数，其中F1是单峰函数，F2-F5是多峰函数，F6-F8是混合函数，其可以是单峰或多峰，F9-F12是复合且多模态的函数。

可以看到在表三显示的数据中，对于CEC 2017大部分的Function，我们测试的结果都已经达到或者接近于最优值了，并且在一些问题上均值也几乎达到了最优值，方差非常小，这意味着NCNES在这些问题上表现出了很稳定的性能。但是，在个别问题上，NCNES的优化显得不足，经过我的观察和推测，这主要是因为$10000\*D$的迭代次数对于目前的更新速度较少，函数值在尚未发生收敛的时候就已经用完了迭代次数，当然这个现象也说明NCNES目前的优化效率有限。另外，在个别Function中，NCNES训练的函数值在未达到最优值时提前收敛，我经过分析，认为这应该是因为后期的多样性指标占比重很低，导致函数丧失了一定的全局最优的搜索能力，从而陷入了局部最优，这个问题应该需要从调整fitness值和diversity值的trade-off来解决，可能需要一个更好的动态的比重系数。

同样的，分析表格五的数据可以得到，首先相当一部分的Function都优化到了最优值，但是其最差值，均值可能还没来得及在迭代结束前达到最优，从而导致较大的方差，这些都表明NCNES的性能与主流的优化算法比较，并没有达到十分理想。另外，注意到$F9$和$F12$，函数在没有达到最优解时提前收敛了，很小的方差表明了函数在每次训练时都陷入了相同的局部最优，这意味着NCNES在多样性指标上，对于全局搜索的比重需要增加，后期需要去优化$\phi$这个平衡fitness和diversity的动态函数。

总结

这段论文总结了一项新提出的数学原理的NCS（Negative Correlation Search）并称为NCNES，通过明确建模和最大化下一代种群的多样性模型（用于探索）和适应性模型（用于开发）。通过针对每个搜索过程进行梯度下降，可以最大化这两种模型。与原始的NCS相比，NCNES具有更清晰的数学解释，展示了如何并行地，负相关地探索解空间，并如何实现。此外，新的NCS还成功解决了原始NCS的两个技术问题。为了评估新NCS的性能，作者提出了一个具体的实例化称为NCNES。NCNES采用了广为人知的NES作为每个子种群的搜索策略。NCNES在不同的Benchmark上表现良好，展现出了算法的优势。.

**CC框架**

进化算法在大多数情况下都是通过在搜索空间中进行随机抽样来工作的在进化算法中常见一个问题就是，当搜索空间显著扩大时，它们的性能通常会急剧下降。为了解决这种大规模问题，人们通常采用的方法之一是CC，Cooperative coevolution。提出CC框架的主要初衷是，遵循分而治之的策略，将一个解分为不同的部分，分别进行优化迭代，目的是尽量使子问题相互独立，相互之间不受影响，从而在解决问题上获得更高的效率。具体的做法可以概括为首先随机产生一个标准的解，然后依次将不同的低维度的部分解补充（覆盖）到标准解上，并且使用前文的评估标准对其进行评估。因此，在采用CC的策略之后，在理想情况下，每个子问题只需要使用进化算法处理自己的部分，同时也相互之间保持独立互不干涉，然后可以使大规模问题得到有效的解决。

**基于不同策略的的CC框架的比较**

提到Cooperative coevolution思想的时候，很容易发现一个主要的困难是在将一个解拆分为多个子问题解的时候，如何评估每一个子问题的解的表现，因为现有的评估函数是基于完整的解给出的，并无法对一个子解做出公平的评估。对于这个问题，传统的CC是在每一轮迭代的时候，将每一个部分解补充上相同的补充向量，将它补全到原始目标函数的维度，然后再做评估，进行优化。这样的做法被称为individual-centric method，在大部分的进化算法环境中，这是一个很好的策略，不仅可以大大减少计算量，提高效率，而且可以省略储存解的相对位置、父代解等信息，这个策略在很多算法上获得了成功。但是在NCNES问题中，由于负相关搜索的特殊性，每一个解可能都处于搜索空间的不同的（局部）最优值上，如果将这些部分解使用相同的补充向量补充，补充进来的维度很可能导致原先的解失去自己在解空间的位置的特殊性。所以对应地，Popovici提出了例如shuffle-and-pair的策略，这样的策略理论上使用不同的补全向量来对子解进行补充，因此在NCNES的情境中，每次我们并不打算让每一个子问题的补充解相同，我使用对应的不同的补充解依次补充到不同的子解上，然后根据梯度更新策略更新这个部分解的高斯概率分布函数。当然，还有其它方式来补全解，或是不需要补全子解的方法，这里不一一叙述。

算法二展示了传统的CC框架下的NCNES

**为了NCNES定制的CC框架**

就如上文提到的，如果使用传统的CC框架，对子解使用相同的补充维度解，会导致所有部分解之间的评估质量差异变得很小，这样就违背了负相关性搜索的初衷，就是要让不同的解尽可能相互远离，这意味着它们在搜索过程中会出现很大的差异，可能会收敛到多个不同的最优解。因此，在NCS中使用相同的补充向量会降低其并行探索能力，并导致错过多个最优解。为了解决这个问题，也可以考虑使用多个不同的补充向量来补充部分解，但这会增加巨大的计算成本，从而降低这个做法的价值。  
因此，我提出了针对于NCNES的CC框架即一种新的协同进化方法，CCNCNES，它的目的是尽可能保留负相关搜索（NCNES）的并行探索性能。与传统的CC不同之处在于，在CCNCNES中，每个子问题的部分解都会被不同的补充向量补充，而不是像传统的CC那样使用相同的补充向量。具体的实施上，每一个部分解都会和本种群父代相同位置的解的高斯分布均值，把均值向量去掉自己的这部分维度之后补充入自己，让自己变为标准维度的解。父代解的高斯分布均值代表了目前对于性能最好的解的期望，因此我们选择使用均值来填补子解的剩余部分来达到完整。因此，CCNCNES实际上是通过顺序优化每个个体的每个子问题来提高NCNES框架下的进化性能。这种策略并没有影响NCNES中不同子解定位于不同子空间的并行探索特性，所以自然也避免了传统CC中一个补充向量导致不同子解越来越近似的情况。

【伪代码3】

根据以上的CC-NCNES伪代码可以看出，相比NCNES，主要的改变就是增加了一个对于第$j$个子问题的循环，将子问题与补充向量合并完整后，对其单独进行NCNES算法的操作。这其中，补充向量$v\_{i,j} = x\_i / x\_ij$，也就是说，对第$j$个子问题的互补向量就是$ x\_i $对于$ x\_ij $的商，这样，选择的补充向量则就是目前认为最适合当前子解的最优期望向量，充分尊重了每一个解的区域特殊性，保留了每一个解的多样性。

实验：

由于本次项目的主要内容是实现NCNES并且加入了Cooperative Coevolution框架，我们设计测试的主要目的是评估NCNES分别在CEC2017的性能和CEC2022的性能，并且由于CC-NCNES是在NCNES的基础上采用了分治的策略，将每一个解个体分为了多个更小的子解，对每一个子解片段单独进行NCNES操作，这种策略实施后理论上减少了同一个解中不同维度的相互依赖性，所以应该可以让每个维度相比之前得到更加明确的优化，从而让算法的收敛效率增加。所以实验部分另外一个内容是在不同的维度数（10，30）上分别运行NCNES和CC-NCNES算法，希望以此发现CC-NCNES优化之后的优势。

根据表4，我将CC-NCNES在CEC2017数据集上运行，可以看到在部分问题上，CC-NCNES表现同样优秀，甚至超过了原本的NCNES。这初步证明了CC-NCNES的有效性，但是由于大部分的问题并没有之前的NCNES表现优秀，这里分析了以下几点原因：CC-NCNES 的优化策略可能更适用于某些特定类型的数据集，而不适用于其他类型。数据集之间的特征差异可能导致算法的表现差异。CC-NCNES 相比于原始的 NCNES 可能引入了更多的复杂性，例如引入了更多的超参数或额外的计算步骤。这可能导致算法更难调优，容易陷入局部最优解。同时CC-NCNES可能更适合高维度问题的求解，所以下文，我使用D=30,30维的相同的CEC2017数据集再分别运行NCNES和CC-NCNES，希望可以得到更明显的区别

相比于 D=10 的情况，我们观察到在 D=30 的数据集上，CC-NCNES 在更多的问题上表现出了优越性。在相同的评估次数下，CC 策略往往能够产生相同甚至更好的解。这表明了 CC-NCNES 在更高维度的问题中具有更强的适应性和优化能力。

一种可能的解释是，在更高维度的问题中，搜索空间更大，传统的优化算法往往容易受到维度灾难的影响，导致性能下降。而 CC-NCNES 利用了协同合作的策略，将高维问题分解为多个低维子问题，并在每个子问题上独立进行优化。这种分解和协作的方式能够更有效地利用计算资源，减轻了维度灾难的影响，从而在更高维度的问题上取得了更好的优化结果。

另外，CC 策略能够在不同子问题之间有效地共享信息，并且通过适当的调整，能够使得优化过程更加平稳和稳健。这种协同合作的优化策略有助于在更高维度的问题上获得更好的性能表现。

总的来说，CC-NCNES 在更高维度的问题上表现出了更好的优化能力，这证明了其在应对维度灾难和复杂问题上的有效性，并为解决实际高维度问题提供了有益的启示。

在Function5中，由于存在距离较远的不同山峰，需要更广泛的搜索空间来覆盖这些山峰。CC策略通过将问题分解为多个子问题，并对每个子问题进行独立的优化，从而在全局范围内实现了广泛的探索。这有助于找到位于不同山峰的局部最优解。在Function9中，存在密集的小山峰，CC策略可以通过精细调整每个子问题的解来实现局部的利用。通过将搜索空间分解为多个子空间，并针对每个子空间进行优化，CC策略可以更有效地探索局部区域并找到局部最优解。

这项论文总结了一项名为NCNES（Negative Correlation Search Enhanced with Negative Entropy Search）的新数学原理，该原理通过明确定义和最大化下一代种群的多样性模型和适应性模型，以实现探索和开发的平衡。NCNES与原始的NCS相比具有更清晰的数学解释，并展示了如何并行和负相关地探索解空间。此外，NCNES还成功解决了原始NCS的两个技术问题，使得算法更加稳健和可靠。

为了评估NCNES的性能，提出了一个具体的实例化方案，并称之为NCNES。NCNES采用了广为人知的NES作为每个子种群的搜索策略，并在不同的Benchmark上进行了测试。结果显示，NCNES在多个Benchmark上表现出良好的性能，并展现了算法的优势。

另外，我还将CC框架引入NCNES中，在高维度的问题中进一步提高了算法的性能。通过将CC与NCNES相结合，算法能够更好地适应高维度问题，并在更广泛的搜索空间中实现更好的探索和开发平衡。这种结合使得NCNES在处理高维度问题时表现出明显的优势，从而提升了算法的实用性和适用范围。