利用遗传算法提高反射面的效率

A. Steklain, [1] **M. Adames 和 F. Ganacim**

数学系，巴拉那联邦科技大学，巴西塞特迪塞滕布罗大道 *3165* 号

电子邮箱：steklain@utfpr.edu.br

摘要：我们提出使用遗传算法来提高一种设备中反射面的效率，该设备的接收器位置不同于经典抛物面天线的接收器位置。通过这一方法，我们证明可以提高 ARAPUCA 光电探测器的效率。

关键词：模拟方法与程序；用于紫外线、可见光和红外光子的光子探测器（真空型）（光电倍增管、混合光电探测器等）；低温探测器

arXiv:2301.08332v2 [physics.ins-det] 7 Feb 2023

[1] 通讯作者。

目录

1 引言 1 2 ARAPUCA模拟 2 3 用于表面优化的遗传算法设计 4 4 结果 6 5 结论 9

1 引言

通常，望远镜和天线的设计会采用反射表面。对于这类设备，传统做法是将表面制成抛物面，使平行电磁波束经抛物面反射后射向焦点，从而最大限度地提高效率。图 1 展示了这一原理。在声学领域也能找到相同的概念，例如采用超材料来产生抛物线相位梯度 [1]。

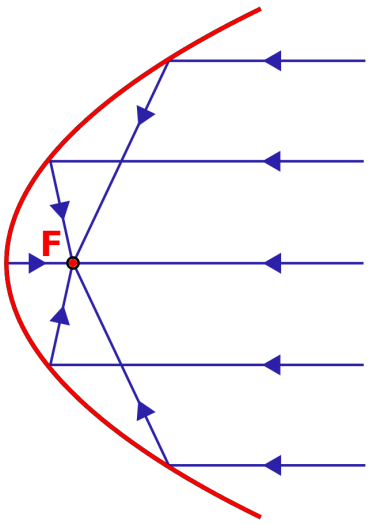

图1。抛物面形状表面的反射。平行光线被反射至焦 s.

近年来，人们提出了采用反射表面的新型装置。特别是，新一代光电探测器（如 ARAPUCA 和 X-ARAPUCA）利用光子陷阱中的反射来提高效率 [2, 3]。这类装置通常在其

结构设计上受到诸多限制，因此即使是离轴抛物面的方案也无法实现。另一方面，光子在被探测到之前，可能会在内部壁面上反射多次。在这种情况下，人们希望采用一种新型表面，使更多反射光汇聚到特定区域（例如光电倍增管所在区域）。尽管没有解析方法可以求得这种表面，但可以通过数值优化算法实现这一目标。

元启发式算法主要用于解决优化问题 [4]。在这些方法中，有一些基于隐喻的算法，其灵感来源于生物或物理现象 [5]。我们还可以根据算法采用单个候选解还是候选解群体，将其分为两类。基于单个解的算法（如模拟退火）可能会陷入局部最优解。另一方面，采用群体的算法能够保持多样性并避免此类问题。基于群体的算法的主要缺点是计算成本较高，因此有必要尽可能减少迭代次数。最为人熟知的基于群体的元启发式算法之一是遗传算法（GA）。

遗传算法背后的理念是被称为进化的生物过程，它结合了自然选择和遗传学 [6, 7]。该算法模仿达尔文进化论，其中适应度最高的个体具有更大的生存和繁殖概率。与自然界中的自然选择不同，个体并不需要强壮或拥有锋利的牙齿才能被选中。我们可以使用任何希望最大化或最小化的量来定义适应度函数，例如光子探测效率。

在本工作中，我们将使用遗传算法寻找能够使 ARAPUCA装置效率最大化的反射表面。我们利用从该表面的离散傅里叶变换中获得的系数对遗传编码进行建模。通过组合这些系数，我们可以找到经过优化的表面，从而提高光探测器的效率。本文的组织结构如下。在第 2 节中，我们介绍 ARAPUCA装置的概念，并描述基于 Python 的模型。在第 3 节中，我们介绍遗传算法中使用的主要程序。在第 4 节中，我们讨论研究结果。最后，在第5 节中，我们给出主要结论。

2 ARAPUCA模拟

ARAPUCA技术带来的主要进步 [2] 在于提高了光探测装置的有效覆盖范围，而无需单纯增加更多的有源光传感器。ARAPUCA 系统结合了无源集光器和有源器件，光子会被困在装置内部，直到被探测到。

我们在图中展示了 ARAPUCA 系统的设计2。其简化的工作原理如下：接收窗口由二向色滤光片和波长转换器（WLS）组成。特定波长范围内的光子可以透过二向色滤光片，并穿过窗口。这些光子被波长转换器吸收，随后以不同的波长重新发射。二向色滤光片会将困在盒体内的新光子完全反射。窗口和内壁会不断反射光子，直到位于其中一面壁上的光传感器（硅光电倍增管）探测到它。

为提高 ARAPUCA 的效率，人们提出了多种改进方案。其中最成功的版本称为 X-ARAPUCA [3]。在该版本中，波长转换发生在置于 ARAPUCA 盒体内的光导中。重新发射的光子向各个

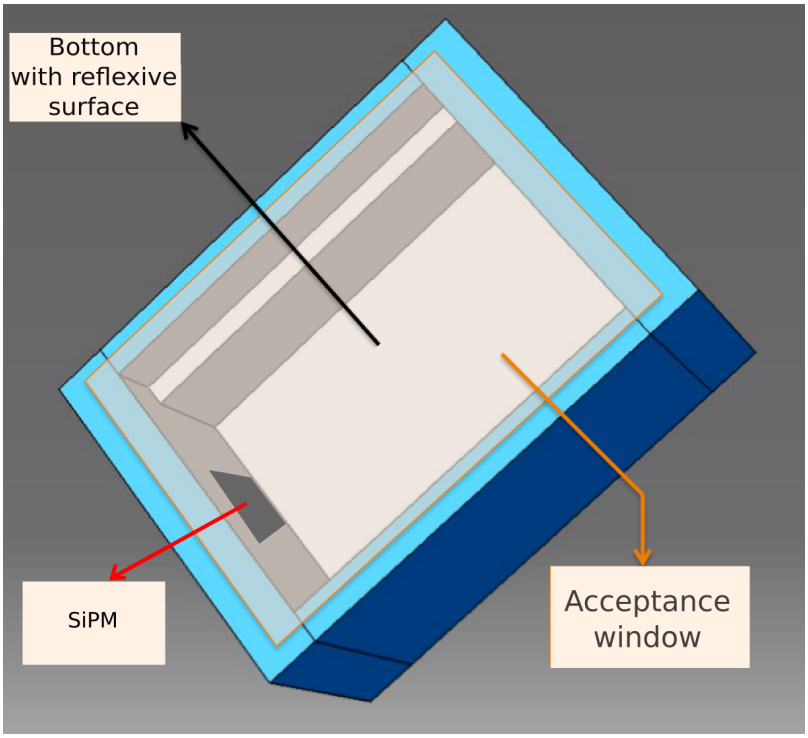

图 2。ARAPUCA装置的设计。入射光子穿过接收窗口后，波长会发生转换。由于该窗口对这一波长具有反射性，光子会被困在盒体内，直到被安装在侧面的有源探测器（硅光电倍增管）探测到。改编自 [2]。

方向传播，并通过全内反射被引导至光传感器。为了提高该系统的效率，人们研究了其他几何设计，但这些设计都在装置底部使用了反射平面壁。由于接收窗口的制造方式，无法对其几何形状进行修改，但可以通过机械加工或 3D 打印来制作底部反射面。本研究旨在优化原始 ARAPUCA 的底部反射面。

在这项工作中，我们结合 Python 和 C++ 射线追踪程序来模拟类 ARAPUCA 装置，因为这种方法简单快捷。不过，为了确认这些结果，最好使用 Geant4 进行包含所有光学性质的完整模拟。

该模拟由一个盒体组成，底部是代表接收窗口的平面，顶部是反射网格表面。在该网格中，节点由 $(x,\ y,\ z)$这些坐标为 $(x,\ y)$ 的点固定，而 $z(x,\ y)$ 允许发生变化，因此该表面可以解释为二元函数的图像。两个表面由四个壁面连接起来，从而封闭盒体，如图 3 所示。光传感器是一个小型正方形，将在盒体内选定，用于计数并阻止撞击该区域的光子。光子在接收窗口中生成，仅在盒体内部传播，并被所有表面点反射，直到到达光传感器或达到最多 50 次反射。我们只统计到达光传感器的光子。

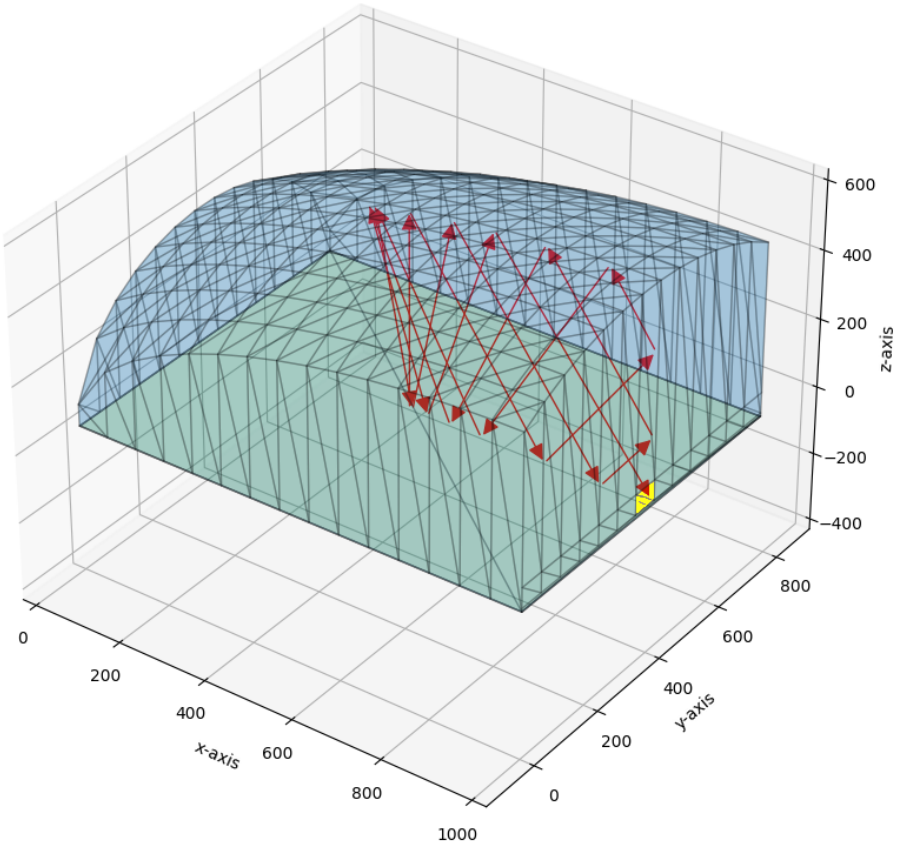

图3。使用 Python 进行 ARAPUCA 模拟。光子以红色箭头表示，从下方的接收窗口（绿色）发出，并由壁面（蓝色）反射。光传感器是位于壁面上的小黄色方块。

3 用于表面优化的遗传算法设计

遗传算法的主要目标是寻找适应度函数的极值。在本研究中，适应度是光电探测器效率，定义为检测到的光子数与生成的光子数之比。生成的光子数是固定的，但检测到的光子数是一个随机变量，因为光子的轨迹是随机生成的。这意味着同一个盒体可能具有不同的效率值。在图 4中，我们展示了同一个盒体的不同效率指标所形成的直方图。如果两个个体的适应度接近，就无法确定哪个更大。为解决这一问题，我们使用统计检验来判断哪个更大。为作出这一判断，我们使用非参数检验，即符号检验 [8]。该检验的形式为

$$T = \frac{1}{M}\sum_{\ell=1}^{M} sgn\left(Z^{(\ell)}\right),$$

其中，$sgn(x) = 1$ 若 $x > 0$ 且 $sgn(x) = -11$ 若 $x < 0$。M 是变量Z 的样本数量，该变量对应于两个不同盒体适应度之间的差异。

遗传算法由若干不同模块组成，必须根据每个问题进行选择和调整。

对于某些系统，这种调整很简单；但对于其他系统，选择则远非易事。遗传算法的主要组成部分包括适应度函数、染色体、初始种群、生存竞争、交配算子、重组和变异。

染色体是算法中最敏感的组成部分，因为它们必须在生成具有不同特征的个体的自由度与方法的收敛性之间取得平衡。尽管显而易见的选择似乎是表面点的 z 坐标，但这种方法并未取得良好效果。主要原因是网格和反射角发生了突变。鉴于表面可以表示为两个变量的函数，我们转而考虑对该表面进行傅里叶分解。染色体是对网格的 z 值进行二维快速傅里叶变换后得到的分量。

初始种群的选择也必须确保必要的遗传多样性，但也可以经过设计，使其表现优于随机生成的个体 [7]。我们没有使用随机个体，而是采用了一组球面截面。这样选择的理由是，从傅里叶分解的角度来看，这些种群仍具有遗传变异性；另一方面，平滑性也能加快算法收敛。

生存竞争负责选择哪些个体将被预先选中并存活下来，从而有机会寻找配偶。尽管一些算法会在“轮盘赌”中采用更高程度的随机化，我们会根据适应度函数的值选取一定比例的种群。

对于交配算子，我们采用 [7] 中所示的策略。适应度较高的个体随机选择配偶，但每个个体被选为配偶的概率与其适应度成正比。这样可以保证适应度较低的个体也有机会被选为配偶，只是概率较低。

所采用的重组方法源自 [9]。它改进了父代的简单交叉

$$parent1 = [m_1, m_2, m_3, ..., m_N]$$
$$parent2 = [d_1, d_2, d_3, ..., d_N]$$

其中，交叉点 α 是一个介于 1 和 N之间的随机整数。随后生成的子代为

$$offspring1 = [m_1, m_2, ..., m_{\alpha-1}, g_1, d_{\alpha+1}..., d_N]$$
$$offspring2 = [d_1, d_2, ..., d_{\alpha-1}, g_2, m_{\alpha+1}..., m_N]$$

其中，在交叉点处，我们对两个父代染色体在位置 α 上的值进行线性组合

$$g_1 = m_\alpha - \beta(m_\alpha - d_\alpha)$$
$$g_2 = d_\alpha + \beta(m_\alpha - d_\alpha)$$

其中，β 是一个介于 0 和 1 之间的随机数。

最后，按照固定概率随机执行变异。变异的强度由另一个随机数决定，其最大值预先设定。

在这项工作中，我们使用由 200 个个体组成的种群。

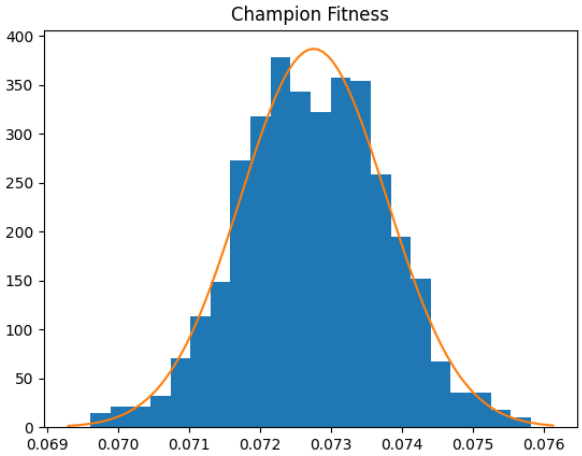

图4。同一盒体不同效率指标生成的直方图。

4 结果

正如我们之前所述，对于平行光束和反射器天线（或望远镜）这一情形，存在一族由抛物面构成的最优解，其中探测器位于顶点处。在这种情况下，效率为 1，即所有反射光子都会被探测器收集。离散网格可能会造成一些光子损失，但效率仍然很高。我们通过将探测器放置在接收窗口中心，模拟了 ARAPUCA模拟中的这种设置，如图 5 所示。

反射器天线是检验所提出的遗传算法的良好对象。我们以一组以探测器为中心、半径各不相同的球面截面作为初始种群。对于这组曲面，最大效率为 0.06。应用遗传算法后，经过 250 代，得到的盒体最大效率达到 0.93。遗传算法应用前后的盒体示例见图6。

对于 ARAPUCA，我们采用一组以探测器为中心的球面作为初始种群。不过，由于探测器位于盒体的一面侧壁上，球面截面看起来与反射器天线的情况不同。

在第一种设置中，我们使用从盒体底部窗口垂直射出的平行光束。在这种情况下，初始种群的最大效率为 0.14。应用遗传算法后，经过 1000 代，最大效率达到 0.52。尽管未能达到更高的效率，但相较于初始设计（半球面片），效率已有显著提升。遗传算法应用前后的盒体示例见图7。

在第二种设置中，我们使用从盒体底部窗口向随机方向发射的光子。在这种情况下，初始最大效率为 0.06；应用遗传算法后，我们发现效率达到了 0.14 的最大值。应用遗传算法前后的盒体示例见图8。值得注意的是，对于随机方向，表现最佳的个体由一个平面反射面构成，这与原始 ARAPUCA 已采用的设计相同。

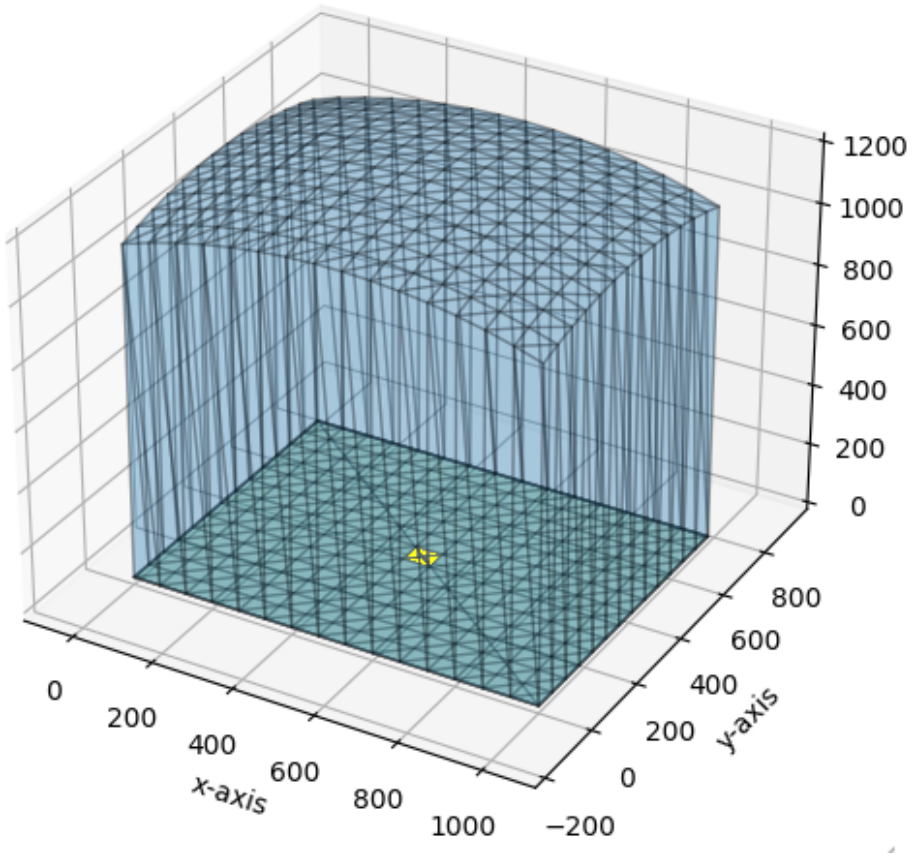

图 5。使用 Python 对抛物面天线进行仿真。光子从接收窗口（绿色）发出，并被壁面（蓝色）反射。光传感器是位于接收窗口中央的小黄色方块。

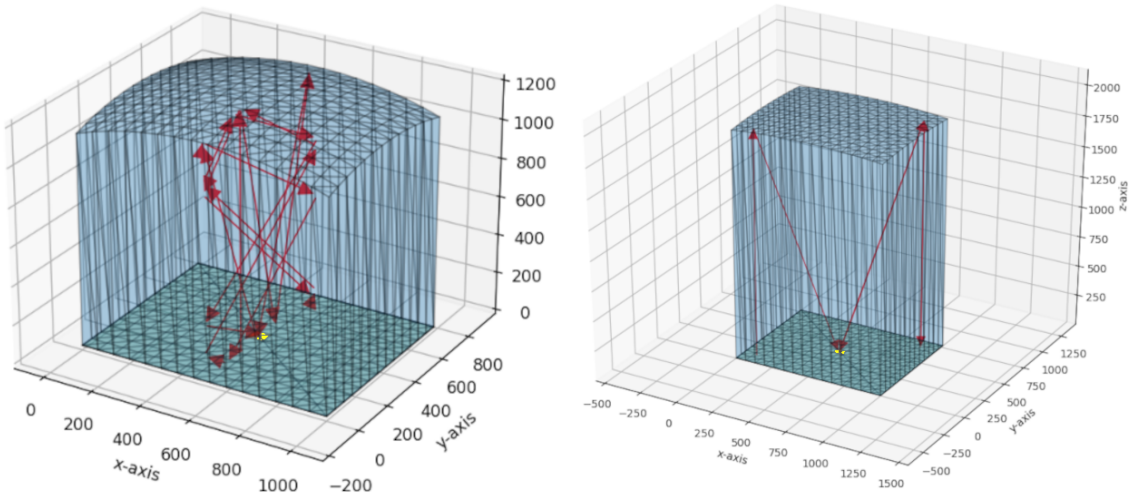

图 6。平行光束下，遗传算法优化前后的反射面。优化前，反射面为半球形，效率为 0.06。经过 250 代后，反射面变为抛物面，效率达到 0.93。

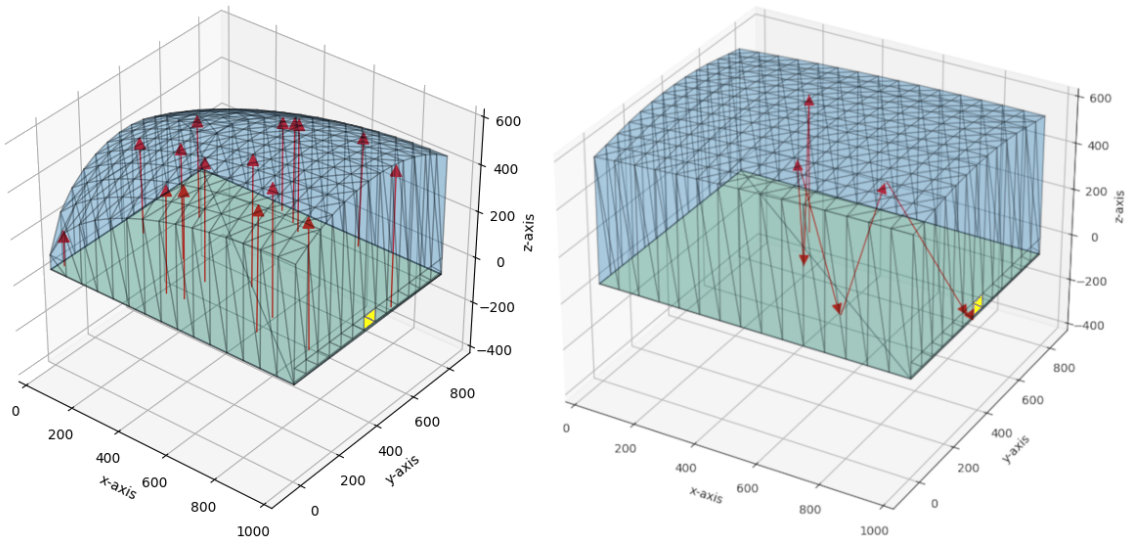

图 7。平行光束下遗传算法优化前后的反射面。优化前，效率为 0.14。经过 1000 代后，效率达到 0.52。

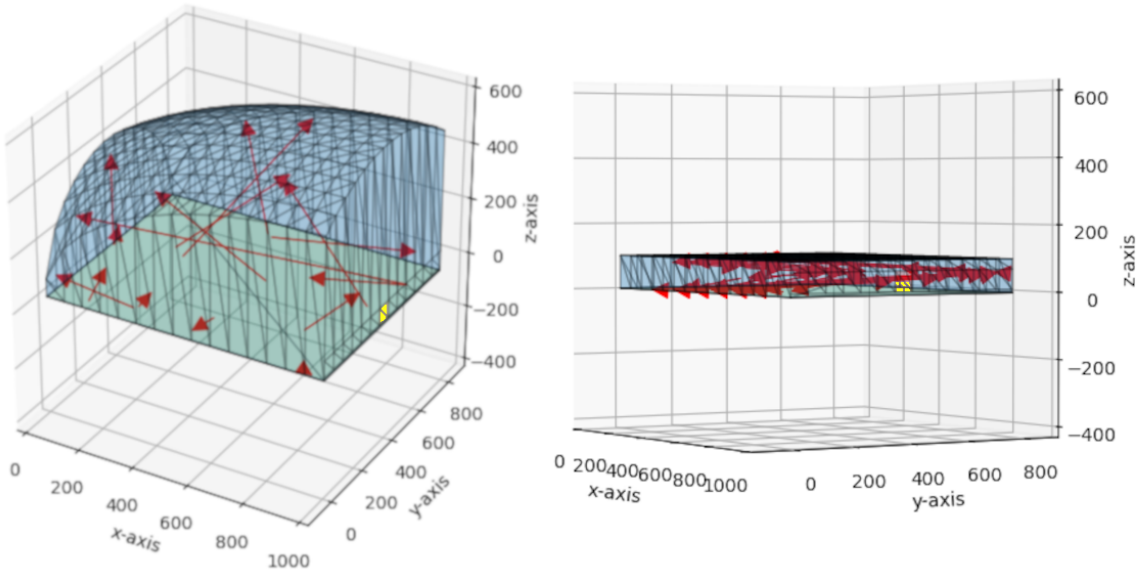

图 8。随机光束下遗传算法优化前后的反射面。优化前，效率为 0.06。经过 1000 代后，效率达到 0.14。在这种情况下，遗传算法无法找到反射面的替代方案，平面是最佳选择。

5 结论

使用遗传算法（GA）优化反射表面时，可以将二维傅里叶分量作为染色体。该方法已在反射天线的优化中证明是成功的。对于 ARAPUCA 系统，该方法改进了平行光束下的反射表面。当光线来自随机方向时，ARAPUCA 目前使用的平面是最佳选择。

在接下来的步骤中，我们将改进 ARAPUCA模拟，使其在物理意义上更加真实。一种可能性是使用 Chroma 或 Geant4 执行光传播。另一个步骤是对 X-ARAPUCA 进行遗传算法（GA）优化。在这种情况下，我们不仅可以考虑反射表面，还可以考虑光导的高度。

致谢

我们感谢 DUNE-BR，尤其是 Ana Machado、Ettore Segreto 和 Gustavo Valdiviesso，感谢他们进行的所有宝贵讨论。

参考文献

[1] *K. Song*、*J. Kim*、*S. Hur*、*J.-H. Kwak*、*S.-H. Lee* 和 *T. Kim*，通过梯度阻碍声学超表面形成的定向反射表面，《科学报告》*6*（*2016*）*32300*。[2] *A.A. Machado* 和 *E. Segreto*，*ARAPUCA*：一种用于液氩闪烁光探测的新装置，《仪器学报》*11*（*2016*）。[3] *A. Machado*、*E. Segreto*、*D. Warner*、*A. Fauth*、*B. Gelli*、*R. Máximo* 等，*X-ARAPUCA*：*ARAPUCA* 装置的改进，《仪器学报》*13*（*2018*）*C04026*[*arXiv:1804.01407v*1]。[4] *M. Abdel-Basset*、*L. Abdel-Fatah* 和 *A.K. Sangaiah*，第*10*章——元启发式算法：综合评述，载于《面向云端多媒体大数据及工程应用的计算智能》，*A.K. Sangaiah*、*M. Sheng* 和 *Z. Zhang* 编，智能数据中心系统，第*185–231*页，学术出版社（*2018*），*DOI*。[5] *S. Katoch*、*S.S. Chauhan* 和 *V. Kumar*，遗传算法综述：过去、现在与未来，《多媒体工具与应用》（*2021*），*10.1007/s11042-020-10139-6*。[6] *D.E. Goldberg*，《搜索、优化与机器学习中的遗传算法》，*Addison-Wesley*，纽约（*1989*）。[7] *M. Kubat*，《机器学习导论》，*Springer* 国际出版公司，*Chan*（*2017*）。[8] *W. Zhai*、*P. Kelly* 和 *W.B. Gong*，含噪适应度的遗传算法，《数学与计算机建模》*23*（*1996*）*131*。[9] *C.F. Toledo*、*L. Oliveira* 和 *P.M. França*，使用具有分层结构种群的遗传算法进行全局优化，《计算与应用数学杂志》*261*（*2014*）*341*。