

1.7 MV 串列加速器调束优化研究

蒋 兑, 李爱玲, 崔保群, 唐 兵, 陈浩南, 王云峰
(中国原子能科学研究院 核物理研究所, 北京 102413)

摘要:为了改进传统的人工调束方法,提升其效率和调优品质,本文引入了差分进化(DE)算法,旨在实现调束过程的智能化。在详细阐述差分进化算法的算法架构基础上,采用Python编程语言,并利用py-EPICS接口与实验物理及工业控制系统(EPICS)建立了稳定的连接。此外,为了方便用户操作与监控,建立了直观的控制系统工作室(CSS)界面,实现了高效的上位控制和实时监测功能。本文基于1.7 MV串列加速器平台对DE算法束流调优的可行性和优化效果进行了在线验证。在实验过程中,不仅对算法的性能进行了全面的评估,还根据实验结果对算法进行了针对性的优化和改进。这些改进措施显著提升了算法的优化能力,使得束流传输效率高达80%。本文不仅展示了DE算法在束流调优中的优异性能,还为智能调束技术的发展提供了新的思路和方法。通过实现调束智能化,有望进一步提高加速器系统的运行效率和稳定性,为相关研究和实践领域提供参考。

关键词:1.7 MV 串列加速器;差分进化算法;束流优化

中图分类号: TL50 文献标志码: A 文章编号: 1000-6931(2025)03-0762-07

doi: [10.7538/yzk.2024.youxian.0529](https://doi.org/10.7538/yzk.2024.youxian.0529)

1.7 MV Tandem Accelerator Beam Tuning Optimization

JIANG Ran, LI Ailing, CUI Baoqun, TANG Bing, CHEN Haonan, WANG Yunfeng
(Department of Nuclear Physics, China Institute of Atomic Energy, Beijing 102413, China)

Abstract: To augment the inherent efficiency and enhance the quality of the conventional manual beam-tuning methodology, this paper presented an innovative approach through the incorporation of a differential evolution (DE) algorithm. Initially, the architectural framework of the DE algorithm was meticulously delineated, serving as the bedrock of the methodological paradigm. The DE algorithm, renowned for its robust optimization capabilities, is implemented utilizing the versatile Python programming language. This implementation leverages Python's computational prowess and inherent flexibility, enabling the development of a sophisticated algorithmic solution. A resilient connection with the experimental physics and industrial control system (EPICS) was established via the pyEPICS interface. This integration facilitates seamless communication and precise control between the advanced DE algorithm and the intricate accelerator system. The pyEPICS interface acted as a conduit, ensuring real-time data exchange and enabling dynamic adjustments to be made based on the algorithm's outputs. Furthermore, to augment user operation and monitoring capabilities, an intuitive control system studio (CSS) interface was devised. This interface empowered efficient upper-level control and real-time monitoring functions, thereby significantly bolstering the usability and practicality of the system. The

CSS interface features a user-friendly graphical user interface (GUI) that allows operators to monitor and adjust parameters in real-time with ease, enhancing the overall user experience and operational efficiency. Using the 1.7 MV tandem accelerator platform as a testbed, rigorous experiments were conducted to ascertain the feasibility and efficacy of the DE algorithm in beam optimization. These experiments were designed to comprehensively evaluate the algorithm's performance under various conditions and constraints. During these trials, this paper not only scrutinized the algorithm's performance but also implemented optimizations and enhancements based on empirical findings. These refinements notably elevate the optimization capabilities of the algorithm, culminating in an impressive beam transfer efficiency of 80%. The methodology encompassed several pivotal steps. Firstly, the DE algorithm using Python was implemented, capitalizing on its robust computational capabilities and inherent flexibility. This implementation allowed for the development of a sophisticated and adaptable algorithmic solution. Subsequently, the algorithm was seamlessly integrated with the EPICS system via the pyEPICS interface, enabling precise control and monitoring of the accelerator beam. The CSS interface was meticulously developed to offer an intuitive and user-friendly graphical interface, facilitating real-time monitoring and adjustment of parameters by operators. The experimental results underscore that the exceptional performance of the DE algorithm in beam tuning. The optimized beam transfer efficiency of 80% constitutes a substantial improvement over traditional manual methods, highlighting the algorithm's efficacy in enhancing beam-tuning processes. Furthermore, the DE algorithm's adaptability and robustness were evident in its proficiency to handle a diverse array of beam conditions and constraints, demonstrating its versatility and practical utility. In conclusion, this study highlights the superior performance of the DE algorithm in beam tuning and proposes a novel approach for the development of intelligent beam-tuning technology. By achieving beam-modulation intelligentization, this paper strives to further enhance the efficiency and stability of accelerator systems. This research not only contributes to the advancement of beam-tuning techniques but also holds considerable promise for related fields of study and practical applications. The findings presented in this paper have the potential to stimulate further research and development in this domain, ultimately culminating in the creation of more efficient and reliable accelerator systems. This work underscores the importance of leveraging advanced algorithmic solutions and robust control systems to enhance the performance and operational efficiency of accelerator facilities.

Key words: 1.7 MV tandem accelerator; differential evolution algorithm; beam optimization

加速器的运行和维护是一个复杂且要求高度精确的过程。随着科技的发展,传统的人工调束方法已经无法满足现代加速器对控制可靠性和实时性的高要求。特别是在面对运行成本高、工作环境复杂以及干扰因素多的情况下,智能调束技术的应用显得尤为重要。1.7 MV 串列加速器是一种能够提供高能粒子束流的设备,广泛应用于材料科学、生物医学、核物理研究等领域。该加速器通过串联多个加速级,逐步增加粒子的能量,最终将粒子加速至所需的能量水平^[1]。

随着加速器技术的不断进步,对加速器的控制精度和稳定性要求越来越高。传统的人工调束

方法不仅耗时耗力,而且难以应对复杂的工作环境和干扰因素。智能调束技术通过自动化和智能化的手段,能够实现对加速器的精确控制,提高运行效率,降低运行成本,并且能够快速响应各种变化,保证实验的顺利进行。本文着重研究 1.7 MV 串列加速器的负氢离子源注入机设计并实现智能调束系统。

1 差分进化算法

差分进化(differential evolution, DE)算法是由 Storn 和 Price 提出的一种基于群体的自适应全局优化算法^[2],该算法主要分为 2 个阶段:一是种群

初始化阶段；二是进化阶段，进化阶段包括变异、交叉、选择等过程，迭代直到满足终止条件。DE 算法作为进化算法家族的一个分支，以其较强的鲁棒性、速度快、执行简单和全局优化能力在各个领域得到了广泛的应用。

1.1 种群初始化

在进化开始之前，需要对 DE 算法中的种群进行初始化，这是一次性操作，不参与迭代过程，其中种群中的个体（又称目标向量）在解空间内初始化。第 i 个个体（又称决策变量）的表达式为：

$$X_i = X_{i,1}, X_{i,2}, X_{i,3}, \dots, X_{i,D} \quad (1)$$

式中， D 为目标函数的维数，即对应优化问题的决策变量的个数^[3]。对于大多数优化问题，解空间都会存在上下边界，分别为：

$$X_{\min} = X_{1,\min}, X_{2,\min}, X_{3,\min}, \dots, X_{D,\min} \quad (2)$$

$$X_{\max} = X_{1,\max}, X_{2,\max}, X_{3,\max}, \dots, X_{D,\max} \quad (3)$$

初始目标向量都应被限制在解空间内。每个目标 i 上界和下界的函数为：

$$x_{i,j} = x_{i,\min} + \text{rand}[0, 1] \times (x_{i,\max} - x_{i,\min}) \quad (4)$$

式中： $\text{rand}[0, 1]$ 为 $[0, 1]$ 范围内的 1 个随机数； j 为第 j 个种群。在最优化过程中，会给出 1 个种群规模（NP），NP 一直保持不变。

在 DE 算法中，评估个体表现的指标被称为适应度函数（fitness function）^[4]。这个函数是判断个体质量的核心标准。适应度函数与所面临的优化问题的目标函数密切相关，其设计对于算法寻找最优解的效率和质量有着决定性的影响。

1.2 差分变异

在 DE 算法的执行流程中，初始种群一旦建立，接下来便进行差分变异步骤。在 DE 算法中，变异过程不是简单的随机扰动，而是通过结合目标向量和其他个体的差分向量来生成新的突变向量^[5]。这个过程可以视为一种基于差异的变异策略，其利用种群中的现有信息来构建新的候选解，从而有助于算法更有效地搜索解空间。

在 DE 算法的差分变异中，目标向量与差分向量的结合生成了突变向量，其数学表达式为：

$$V_i^G = X_i^G + F \cdot (X_{r1}^G - X_{r2}^G) \quad (5)$$

式中： $i, r1$ 和 $r2$ 为在 $[1, NP]$ 之间随机选择的不同整数； V_i^G 为由突变产生的突变个体； F 为缩放因子，用于控制差分向量的影响程度^[5]； X_i^G 为第 G 次迭代的第 i 个个体。

总结来说，DE 算法的差分变异过程通过种群内部的相互作用来生成新的解，而不是依赖于随机扰动，这使得 DE 算法在处理复杂优化问题时表现出了良好的性能和鲁棒性。

1.3 交叉和选择

突变后，突变和目标载体的成分根据交叉率（CR）进行交换，以产生后代，在 DE 算法中也称为试验载体。有两种常用的交叉策略，包括二项交叉和指数交叉^[5]。根据二项交叉产生的试验向量为：

$$U_{i,j}^G = \begin{cases} V_{i,j}^G & \text{if } \text{rand}[0, 1] \leq CR \text{ or } j = j_{\text{rand}} \\ X_{i,j}^G & \text{otherwise} \end{cases} \quad (6)$$

式中： $U_{i,j}^G$ 为通过目标向量 $X_{i,j}^G$ 和突变向量 $V_{i,j}^G$ 进行交叉操作生成的试验向量； j_{rand} 为在 $[1, D]$ 内随机选择的整数，以确保供体向量中的至少 1 个维度传递给试验向量^[5]。通过这种方式，DE 算法能够有效地探索解空间，并保持种群的多样性，同时避免了遗传算法中可能出现的早熟收敛问题。

DE 算法中的选择过程对于确定哪个载体，无论是试验载体还是目标载体，将会被保留到下一代是至关重要的。在选择之前，所有的测试向量都应该通过目标函数进行评估，以获得适应度。适应度较差的向量被丢弃，而适应度较好的向量被保留。选择操作为：

$$X_i^{G+1} = \begin{cases} U_i^G & \text{if } f(U_i^G) \leq f(X_i^G) \\ X_i^G & \text{otherwise} \end{cases} \quad (7)$$

式中： $f(U_i^G)$ 和 $f(X_i^G)$ 分别为试验向量和目标向量的适应度； X_i^{G+1} 为第 $G+1$ 次迭代的第 i 个个体。

1.4 算法流程

DE 算法的整体流程如下：1) 随机生成初代种群，迭代次数为 0，即 $g=0$ ；2) 计算每个种群对应的初始值向量；3) 判断是否达到终止条件或达到最大迭代数，如果满足，则程序终止并且输出最佳解向量，否则继续执行步骤 4；4) 进行差分变异和交叉操作，并计算种群中的适应度值，最后进行选择操作，得到新的种群；5) $g=g+1$ ，以新的种群继续进行步骤 3 的操作。DE 算法流程图如图 1 所示。

2 在线实验

2.1 实验流程

由于 1.7 MV 串列加速器高压加速段及高能

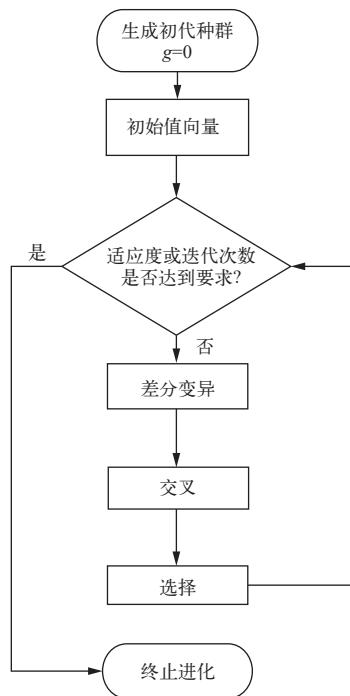


图 1 DE 算法流程图

Fig. 1 DE algorithm flowchart

束流线尚在调试阶段,为了测试本文采用的优化方法,对串列加速器低能束流线进行在线实验测试,可调的器件包括:1台灯丝加热电源、1台弧放电源、3台高压电源(分别用于引出电压、二次加速电压和单透镜电压)、1台磁铁稳流电源和2个法拉第筒(分别记为F1和F2)^[1],串列加速器低能束流线如图2所示。

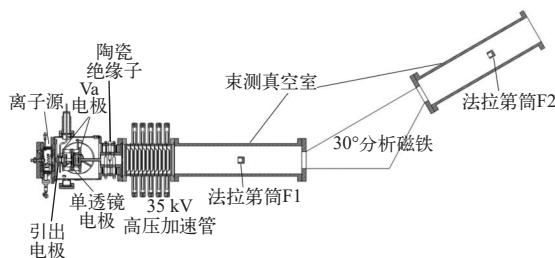


图 2 串列加速器低能束流线

Fig. 2 Low energy beam line for tandem accelerator

本文采用直接在线优化方法,目标函数来自于法拉第筒测得的束流信息,由皮安表进行采集。引出电极、Va电极、单透镜电极和30°分析磁铁的电压构成决策变量。根据设备最初设计时设定的运行参数,确定变量的参数范围,即确定解空间的边界。适应度为传输效率,即F2与F1的比值。

本文采用Python语言实现DE算法的程序,由于在控制系统下,所有的决策变量均被封装成了实验物理及工业控制系统(experimental physics and industrial control system, EPICS)的过程变量(process variable, PV),以pyEPICS作为访问控制系统PV的工具^[3]。同时,编写控制系统工作室(control system studio, CSS)界面便于观测调束过程中各决策变量的变化以及优化后得到的束流信息^[6]。此外,还可以在CSS界面实现对优化程序参数值的设定,更加方便快捷,优化程序与控制系统的集成结构示意图如图3所示,其中IOC为输入/输出控制器,PS_n为第n个电源。

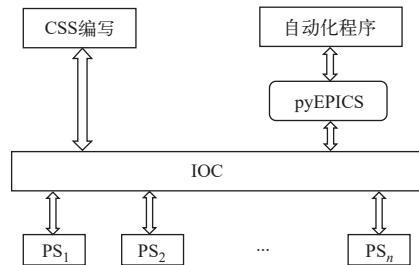


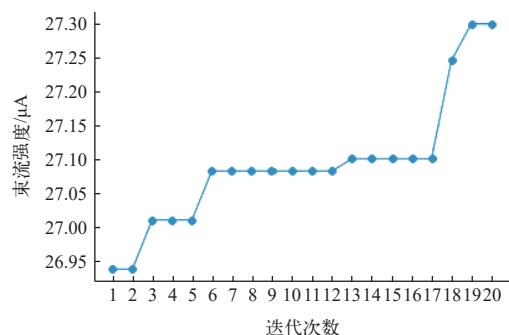
图 3 优化程序与控制系统的集成结构

Fig. 3 Integrated structure of optimization program and control system

2.2 实验结果与分析

2.2.1 H⁻的DE算法寻优 初始实验中优化变量为4台电源,即D=4、NP=5、F=0.8、CR=0.7。优化目标为:调束优化时间达到30 min以内,传输效率达到80%。在CSS界面设置好各个变量的取值范围,对H⁻进行了DE算法调束测试,得到的束流调优曲线如图4所示,本文采集的H⁻离子束的数据均为绝对值。

在同样的实验条件下,传统的人工调束时间

图 4 H⁻束流调优曲线Fig. 4 Beam optimization curve of H⁻

约为3~4 h, 传输效率约为66%。对于DE算法调节的H⁻(43 keV)离子束, 当F1的电流增加至42.4 μA时, 由图4可看出, F2的电流可提升至27.3 μA, 此时的传输效率可以达到62%, 调束时间约为17 min。

2.2.2 DE算法的优化 由2.2.1节可知, H⁻(43 keV)离子束的寻优效果与传统的人工调束相比, 调束优化时间大大缩短, 但束流的传输效率并没有明显增强。因此, 本文要对DE算法的性能进行优化。根据初始实验数据可以看出, DE算法主要的控制参数包括种群规模、缩放因子和交叉率。

种群规模主要反映算法中种群信息量的大小^[7], 种群规模越大包含的种群信息越丰富, 但是会增加计算量, 进而提高迭代时间; 反之, 种群多样性受到限制, 不利于算法求得全局最优解, 甚至会导致搜索停滞。一般情况下, 对于小群体目标优化, 种群规模取值范围在D~4D之间, 且要大于4, 否则会影响整个算法的迭代。

在本文中, 由于设备还在调试阶段, 主要围绕4台电源(D=4)进行实验, 决策变量较少, 重点对比了种群规模为9和15两种情况下的束流调优效果, 得到了如图5所示的束流调优曲线。从图5中可以观察到, NP=15的曲线在迭代结束时并没有明显的收敛, 由此猜测20次迭代并没有找到最优值, 增加迭代次数至50次, 得到了如图6所示的束流调优曲线。与NP=5时相比, 种群规模增大后, 最优值得到了显著提升。

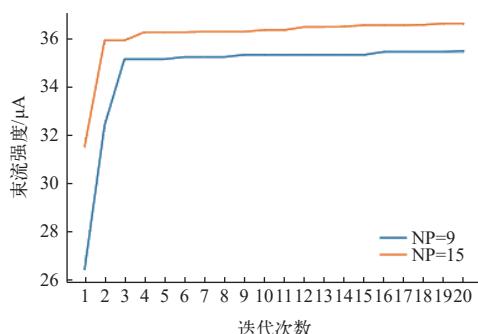


图5 迭代20次时的束流调优曲线
Fig. 5 Beam optimization curve at 20 iterations

在实验过程中, 详细记录了不同种群规模的调束优化时间, 主要包含监测变量的采样率(约为300 ms)、测量延时(约为100 ms)、控制变量的数据刷新率(毫秒级)、PLC数据扫描时间(约为

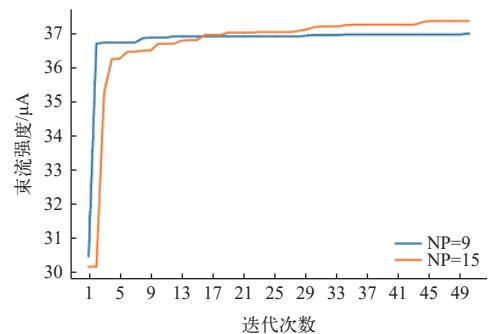


图6 迭代50次时的束流调优曲线
Fig. 6 Beam optimization curve at 50 iterations

几十毫秒)、IOC服务器响应时间(约为0.5 s)以及程序迭代时间(主要耗时)。具体而言, 当种群规模为9时, 整个种群的迭代20次的调束优化时间为22 min, 迭代50次的调束优化时间为58 min; 当种群规模增至15时, 迭代20次的时间延长至38 min, 迭代50次的时间延长至80 min。值得注意的是, 种群规模从9增加到15, 寻优得到的最优值从35.5 μA提升至36.6 μA, 绝对差异约为1 μA。尽管这一差异相对于整体优化范围而言较小, 但对于低能束流线来说, 微小的变化也可能对实验结果和束流线的性能产生显著影响, 因此并不能被忽略。

综合对比调束优化时间和寻优效果, 基于优化目标, 可以得出结论: 在本文中, 种群规模为9时, 能够在保证寻优效果的同时, 更有效地节约计算资源。因此, 从实际应用的角度出发, NP=9的效果更佳。

交叉率主要反映的是在交叉的过程中, 子代与父代、中间变异体之间交换信息量的大小程度^[8]。交叉率越大, 信息量交换的程度越大。反之, 如果交叉率偏小, 将会使种群的多样性快速减小, 不利于全局寻优。本文主要取交叉率为0.6、0.7、0.8, 并对其寻优效果进行比较, 如图7所示。从图7中可明显看出, CR=0.7时效果最佳。

相对于交叉率, 缩放因子对算法性能的影响更大, 缩放因子主要影响算法的全局寻优能力^[2]。缩放因子越小, 算法对局部的搜索能力更好, 缩放因子越大, 算法越能跳出局部极小点, 但是收敛速度会变慢。此外, 缩放因子还影响种群的多样性。为了探讨缩放因子的不同取值对寻优效果的影响, 分别选取了缩放因子为0.7、0.8和0.9进行对比实验, 实验结果如图8所示。

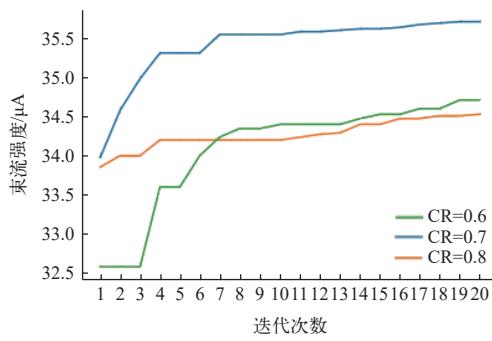


图7 不同交叉率时的束流调优曲线

Fig. 7 Beam optimization curve of different CR

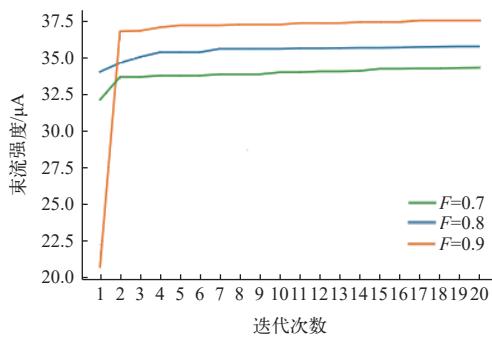


图8 不同缩放因子时的束流调优曲线

Fig. 8 Beam optimization curve of different F

通过仔细观察和分析实验数据发现,当 $F=0.8$ 时,寻优过程中得到的最优值相对较大。更值得一提的是,在 $F=0.8$ 的条件下,算法的收敛速度也表现出明显的优势,相较于其他两个取值($F=0.7$ 和 $F=0.9$),其更快地接近了最优解。

综合考量最优值的大小和收敛速度的快慢,在本文实验条件下,当缩放因子取值为0.8时,寻优效果达到最佳。

由各个参数的实验对比可知,当DE算法的参数为 $NP=9$ 、 $CR=0.7$ 、 $F=0.8$ 时寻优效果较好,且满足优化目标。根据这组参数进行了20次和50次迭代,得到的实验结果如图9所示。调束优化时间分别约为22 min和58 min,由图9中数据可看出,20次迭代的最优值与50次迭代的最优值的差别较小。因此,在目前决策变量不变的前提下,迭代次数设置为20次时,DE算法就能得到很好的寻优效果,工作效率较高。

3 结论

本文将DE算法引入到调束优化过程中,旨在通过智能算法提升束流传输效果。为了验证该算

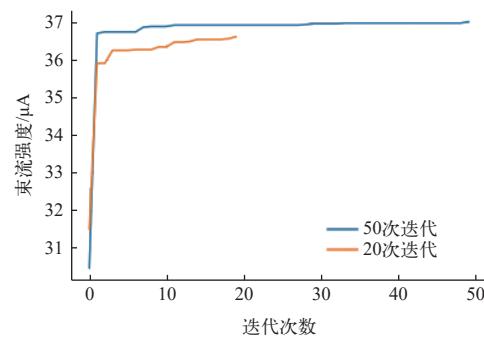


图9 20次和50次迭代寻优效果对比

Fig. 9 Comparison of optimization result between 20 and 50 iterations

法的可行性和优化效果,在1.7 MV串列加速器平台上进行了在线实验验证。在实验过程中,详细记录了算法的运行情况,并根据实验结果对其性能进行了针对性的提升。

实验结果表明,通过引入DE算法进行优化, $F2$ 处的束流强度得到了显著提升,最高可达到 $37.5 \mu\text{A}$ 。与此同时,传统的人工调束需要花费 $3\sim4$ h才能达到66%的传输效率,相比之下,在没有人工干预的情况下,束流的传输效率大幅提高,达到了80%,且调束优化时间大大缩短,不管串列头部高压多大,时间都可以缩短至22 min左右。这一成果不仅验证了DE算法在调束优化中的有效性,也展示了该算法在提升束流传输效率方面的巨大潜力。

此外,为了验证DE算法的稳定可靠性,在超导直线加速器上对该算法进行了在线验证,并取得了较好的优化效果。

本文仅呈现了串列加速器低能束流线的调束工作,选取的样本参数(即决策变量)较少,这在一定程度上限制了优化空间的探索,根据理论计算对被调电源设置了调节范围,并通过实验验证达到了该装置最初设定的目标优化效果,得到的最佳解可能为局部最佳解。随着串列加速器束流线的完善,决策变量会更加丰富,后续会对该装置继续优化寻找全局最佳解,并基于决策变量进行进一步探究,如决策变量与算法各参数的关系、决策变量与调束效率的关系等,继续完善整个1.7 MV串列加速器束流线的调束数据。

本文研究为DE算法在调束优化领域的应用提供了有力的实验支持,同时也为后续相关研究提供了新的思路和方法。

参考文献:

[1] NAAB F U, TOADER O F, WAS G S. Ion beam transport simulations for the 1.7 MV tandem accelerator at the Michigan Ion Beam Laboratory[J]. *Physics Procedia*, 2015, 66: 632-640.

[2] 王兴锋. 离散差分进化算法与工程应用研究[D]. 太原: 太原科技大学, 2016.

[3] 时英智, 高梅, 贾文红, 等. 基于 EPICS 和差分进化算法的同步辐射光束线智能调束系统[J]. 核技术, 2020, 43(5): 050101.

SHI Yingzhi, GAO Mei, JIA Wenhong, et al. Intelligent commissioning system based on EPICS and differential evolution algorithm for synchrotron radiation beamline[J]. *Nuclear Techniques*, 2020, 43(5): 050101(in Chinese).

[4] 时英智. 基于差分进化算法的光束线智能调束系统[D]. 上海: 中国科学院大学(中国科学院上海应用物理研究所), 2020.

[5] YU L, REN C, MENG Z. A surrogate-assisted differential evolution with fitness-independent parameter adaptation for high-dimensional expensive optimization[J]. *Information Sciences*, 2024, 662: 120246.

[6] 常建军. 320 kV 实验平台 EPICS 应用及调束优化研究[D]. 兰州: 中国科学院大学(中国科学院近代物理研究所), 2019.

[7] KUMAR P, ALI M. Improved differential evolution algorithm guided by best and worst positions exploration dynamics[J]. *Biomimetics*, 2024, 9(2): 119.

[8] 侯现钦, 王良明, 傅健. 差分进化智能算法在高旋弹气动辨识中的应用[J]. 弹箭与制导学报, 2020, 40(3): 103-107.

HOU Xianqin, WANG Liangming, FU Jian. Application of different evolution intelligent algorithm in high-vibration aerodynamic identification[J]. *Journal of Projectiles, Rockets, Missiles and Guidance*, 2020, 40(3): 103-107(in Chinese).