

文章编号: 2095-2163(2021)07-0180-06

中图分类号: TP391

文献标志码: A

基于双重改进的鲸鱼优化算法

陈楠, 高建瓴, 喻明毫, 白羽飞, 胡承刚

(贵州大学 大数据与信息工程学院, 贵阳 550025)

摘要: 传统的鲸鱼优化算法(WOA)容易陷入局部最优以及收敛速度慢,针对此问题进行研究,提出了一种改进的鲸鱼优化算法,改进算法首先用非线性收敛因子替换原本使用的收敛因子,改进后的非线性收敛因子可以有效利用在算法中以弥补该算法在计算过程中全局探索与局部开发能力中的缺陷,并且可以加快算法收敛速度;然后在鲸鱼位置更新公式中加入了自适应权重,该策略可以改善算法的寻优精度以及进一步提高收敛速度;最后,在固定参数和不同维度的8个基准测试函数上进行了实验,结果表明,改进后的算法在寻找最优位置的精度和收敛速度对比于传统的鲸鱼算法和其他智能优化算法均有着显著的提高,具有更好的优化效果。

关键词: 鲸鱼优化算法; 自适应权重; 收敛因子; 寻优精度; 收敛速度

A hybrid improved whale optimization algorithm

CHEN Nan, GAO Jianling, YU Minghao, BAI Yufei, HU Chenggang

(College of Big Data and Information Engineering, Guizhou University, Guiyang 550025, China)

【Abstract】 The traditional whale optimization algorithm (WOA) is easy to fall into local optimum and has slow convergence speed. Aiming at this problem, an improved WOA is proposed. Firstly, the nonlinear convergence factor is used to replace the original one. The improved non-linear convergence factor is helpful to make up for the defects of the algorithm in global exploration and local development ability, and can improve the efficiency of the algorithm, meanwhile speed up the convergence speed of the algorithm; then, the adaptive weight is added to the whale position update formula, which can improve the optimization accuracy and further improve the convergence speed of the algorithm; finally, experiments are carried out on eight benchmark functions with fixed parameters and different dimensions, and the results show that the accuracy and convergence speed of the improved algorithm in finding the optimal position are significantly improved compared with the traditional whale algorithm and other intelligent optimization algorithms, and it has better optimization effect.

【Key words】 whale optimization algorithm; adaptive weight; convergence factor; optimization accuracy; convergence speed

0 引言

鲸鱼优化算法(Whale Optimization Algorithm, WOA)是位于澳大利亚的格里菲斯大学中的Mirjalili等人^[1]于2016年提出的一种新型的群体智能优化方法,其想法来自于座头鲸在海洋中独特的捕捉食物的行为。该算法通过鲸鱼包围猎物、泡泡攻击猎物这些过程来实现优化搜索。该算法的原理相对简单、操作起来也比较容易、容易实现、需要调整的参数比较少、一级鲁棒性较强等优点。在函数优化方面,相比较于粒子群(PSO)、差分进化(DE)和引力搜索(GSA)、萤火虫(FFA)等算法,WOA算法的稳定性和求解精度也明显较高,同时,该算法也越来越多被应用在实际工程中, Kumar等人将

WOA应用于全球MPP光伏系统的跟踪系统中,实现了能源的有效利用。Mohamed等人将WOA应用于多阈值图像分割的问题中,实验结果表明,WOA算法的性能在大多数情况下都优于其他对比算法。

然而,和其他元启发式算法大概一致,传统的鲸鱼优化算法进行计算时依旧存在一些缺陷,例如求出最优解的精度相对较低、算法计算起来要进行收敛的速度也很缓慢、收敛后期算法容易停滞不进行计算、还容易陷入局部最优解、收敛性证明以及自身的参数选择准则等。Pradeep等人将自适应策略引入到WOA中,并将这个改进后的算法应用于10个经典的函数优化问题中,仿真结果比较得出,所提出的改进的自适应WOA算法在收敛速度和精度上均比原算法更佳。Oliva等人提出了混沌映射鲸鱼

作者简介: 陈楠(1997-),女,硕士研究生,主要研究方向:信息与通信工程、深度学习;高建瓴(1969-),女,硕士,副教授,主要研究方向:数据分析、数据库应用;喻明毫(1997-),男,硕士研究生,主要研究方向:信息与通信工程、深度学习;白羽飞(1995-),男,硕士研究生,主要研究方向:信息与通信工程、文本处理;胡承刚(1996-),男,硕士研究生,主要研究方向:自然语言处理。

通讯作者: 高建瓴 Email:454965711@qq.com

收稿日期: 2021-04-09

优化算法(CWOA),并且在太阳能电池和其光伏组件或者面板的参数预测中使用该算法,实验结果对比表明,所提出的混沌映射算法在解决原先问题中的预测精度以及鲁棒性上都有比较好的效果。因此,针对在传统的WOA算法中一直存在的优化问题,本文提出了基于双重策略改进的鲸鱼优化算法(Improved whale optimization algorithm based on dual strategy, CWOA)。改进算法首先用非线性收敛因子替换原本使用的收敛因子,改进后的非线性收敛因子可以有效利用在算法中以弥补该算法在计算过程中全局探索与局部开发能力中的缺陷,并且可以使算法收敛速度加快;然后在鲸鱼位置更新公式中加入了自适应权重^[2],该策略可以改善算法的寻优精度以及进一步提高收敛速度。仿真结果表明,该算法具有较高的优化精度和较快的收敛速度。

1 鲸鱼优化算法

受到鲸鱼特殊捕食行为的启发,Mirjalili 等人于 2016 年提出的一种新型的群体智能优化方法—鲸鱼优化算法(WOA)^[3],该算法的原理在于模拟座头鲸进行泡泡网捕食时的过程,如图 1 所示。鲸鱼优化算法分为以下的 3 个部分,分别是:对猎物进行包围、泡泡攻击猎物以及搜寻猎物。



图 1 泡泡网捕食

Fig. 1 Bubble-net predation

1.1 对猎物进行包围

首先是通过向上螺旋靠近目标,并逐渐缩小包围范围,最终到达目标鱼群的位置的方法,即为找到问题的最优解。假设鲸鱼种群的所有规模为 N ,需要进行最优解问题求解所在空间的维度为 D ,则需要求解第 i 只鲸鱼在 D 维空间所求解的对应的位置为 $X_i = (x_i^1, x_i^2, \dots, x_i^d)$, $i = 1, 2, 3, \dots, N$, 则最优鲸鱼的位置对应于问题的最优解^[4]。

在捕食过程中,鲸鱼首先会观察并识别猎物所在的位置,然后对其进行包围。在鲸鱼优化算法中,假

设问题最佳的结果就是猎物所处的位置,当猎物被定位好时,其他的鲸鱼也会游向该鲸鱼的位置。反之,在鲸鱼优化算法中,个体和最优解(猎物)之间的距离需要第一个求解。此时需用到的数学公式为:

$$\vec{D} = |\vec{C} \cdot \vec{X}^*(t) - \vec{X}(t)| \tag{1}$$

其中, t 为当前已进行迭代过的次数; $\vec{X}(t)$ 为各个体位置向量; $\vec{X}^*(t)$ 为目前求出来的最优解,即猎物所在位置;常数 \vec{C} 为摆动因子,可由下列式子计算得出:

$$\vec{C} = 2 \cdot \vec{r} \tag{2}$$

鲸鱼位置更新公式为:

$$\vec{X}(t+1) = \vec{X}^*(t) - \vec{A} \cdot \vec{D} \tag{3}$$

其中, \vec{A} 和 \vec{D} 为矩阵系数表达式为:

$$\begin{cases} \vec{A} = 2a \cdot \vec{r}_1 - a \\ \vec{D} = 2 \cdot \vec{r}_2 \\ a = 2 - \frac{2t}{T_{\max}} \end{cases} \tag{4}$$

其中, \vec{r} 为 $[0, 1]$ 之间的随机数, a 随着迭代次数的变化而变化,迭代次数增加时, a 也从 2 到 0 相应地在线性递减, T_{\max} 表示迭代次数最大值^[5]。

1.2 泡泡网攻击

鲸鱼的泡泡网攻击有 2 种方式。一种是收缩包围捕食,另一种是螺旋吐气泡捕食。对此拟做研究分述如下。

(1) 收缩包围捕食。此方式中鲸鱼的位置是通过式(4)来获得的,通过其中收敛因子 \vec{a} 来进行求解。

(2) 螺旋式吐气泡捕食。第一步先计算鲸鱼各个个体与当前最佳位置(当前最优解)之间的距离,然后模拟鲸鱼—在捕食时使用螺旋向上的方式进行计算,其公式为:

$$\vec{X}(t+1) = \vec{D} \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) \tag{5}$$

其中, $\vec{D} = |\vec{X}^*(t) - \vec{X}(t)|$ 表示第 i 只鲸鱼和当前最佳位置(最优解)之间的距离; b 是用来对螺旋形式的常量系数进行限定; l 为 $[-1, 1]$ 之间产生的一个随机值。需要特别注意的是,在鲸鱼使用螺旋向上的形式对猎物进行包围时,还需要对包围的圈子进行收缩。因此,为了实现该同步模型,选择相同概率 p 来对最优解进行收缩包围和螺旋更新^[6],其公式如下:

$$\vec{X}(t+1) = \begin{cases} \vec{X}^*(t) - \vec{A} \cdot \vec{D} & \text{if } p < 0.5 \\ \vec{D} \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) & \text{if } p \geq 0.5 \end{cases} \quad (6)$$

其中, p 为 $[0,1]$ 之间的随机数。

1.3 搜寻猎物

当 $|\vec{A}| > 1$ 的时候, 鲸鱼将针对猎物进行随机的搜索, 进行全局范围的搜索^[7], 就是为了避免该算法在计算过程中陷入局部最优^[5], 该阶段的数学表达式如下:

$$\vec{D} = |\vec{C} \cdot \vec{X}_{rand} - \vec{X}(t)| \quad (7)$$

$$\vec{X}(t+1) = \vec{X}_{rand} - \vec{A} \cdot \vec{D} \quad (8)$$

其中, \vec{X}_{rand} 为当前计算种群中随机挑选的一只鲸鱼的位置。

2 双重改进鲸鱼优化算法

鲸鱼优化算法在计算时收敛速度比较慢、还容易在计算过程中陷入局部最优以及容易早熟收敛, 针对这些问题, 提出利用非线性收敛因子和自适应权重对鲸鱼优化算法进行改进。本文算法流程如图2。

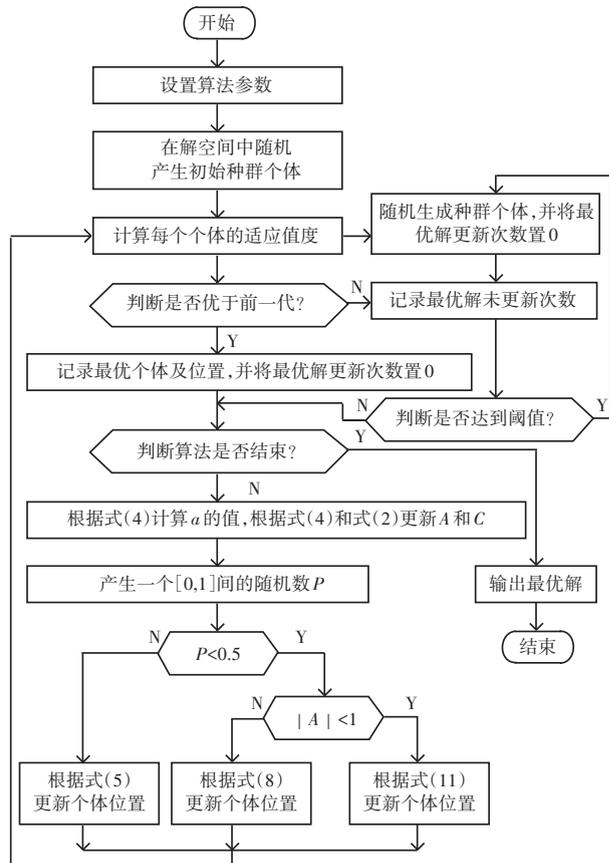


图2 算法流程图

Fig. 2 Algorithm flow chart

2.1 非线性收敛因子

在寻找最优解的过程中, 鲸鱼优化算法和其他群体智能算法一样, 会出现全局探索能力和局部开发能力不平衡的现象^[8]。对于鲸鱼优化算法进行分析, 从结果可知, 收敛因子 \vec{a} 针对全局探索能力以及局部开发能力是特别重要的一部分, 在种群中各个个体之间的位置发生变化也与这个收敛因子有关, 但是在经典的鲸鱼优化算法中收敛因子 \vec{a} 随着迭代次数的一次次增加线性地从 2 递减到 0, 这也使得该算法的迭代的速度变得相对比较缓慢; 同时算法在进行局部求解最优时, 搜索答案只能靠近局部最优解, 而进行局部寻优不能有更好的效果。针对这个问题, 本文提出了一种非线性收敛因子^[9]。具体公式如下:

$$\vec{a} = 1 - \sin(t * (\frac{2}{Max_iter})) \quad (9)$$

其中, Max_iter 为最大迭代次数, t 为当前迭代次数。

2.2 自适应权重策略

鲸鱼优化算法在计算后期进行局部位置开发时容易陷入局部最优, 也特别容易出现早熟收敛的现象, 为此, 提出一种自适应权重策略, 为了使算法能够保持种群的多样性并且能够及时跳出局部最优。

自适应权重策略函数公式如下:

$$w = 1 - \frac{e^{\frac{t}{Max_iter}} - 1}{e - 1} \quad (10)$$

$$\vec{X}(t+1) = w \cdot \vec{X}^*(t) - \vec{A} \cdot \vec{D} \quad (11)$$

$$\vec{X}(t+1) = \vec{D} \cdot e^{bl} \cdot \cos(2\pi l) + w \cdot \vec{X}^*(t) \quad (12)$$

每个个体都要经过对猎物进行包围、泡泡网攻击、搜寻最后的猎物三个阶段, 当个体在对猎物进行包围阶段或者泡泡网攻击阶段时采用自适应权重策略去更新位置, 取该个体进行变化后的最优的位置, 为了让种群的收敛变得更快, 也可以防止种群陷入局部最优的效果变得更好, 使算法具有更好的寻优效果。

3 实验结果与分析

3.1 实验环境和测试函数

本文的仿真实验是基于 Windows 10(x64) 操作系统, Intel^(R) Core^(TM) i5-7500 CPU 3.40 GHz 主频, RAM 8 G 内存。编译环境采用 python3.6。为验证本文新提出的 CWOA 的寻求最优解的性能, 参照文

献选取 8 个基准测试函数进行数值仿真实验, 具体的测试函数见表 1, 其中 $F_1 \sim F_4$ 为单模态函数, 用

来测试算法局部寻优能力; $F_5 \sim F_8$ 表示复杂的非线性多模态函数, 用来测试算法的全局寻优。

表 1 测试函数

Tab. 1 Test function

测试函数	范围	理论最优
$F_1 = \sum_{i=1}^n x_i^2$	$[-100, 100]$	0
$F_2 = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	$[-10, 10]$	0
$F_3 = \sum_{i=1}^n (\sum_{j=1}^n x_j)^2$	$[-100, 100]$	0
$F_4 = \max\{ x_i , 1 \leq i \leq n\}$	$[-100, 100]$	0
$F_5 = \sum_{i=1}^n ix_i^4 + \text{random}[0, 1)$	$[-1.28, 1.28]$	0
$F_6 = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	$[-500, 500]$	-4 187.982 9x5
$F_7 = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	$[-32, 32]$	0
$F_8 = \frac{1}{4000}\sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600, 600]$	0

3.2 参数设置与实验结果分析

为了保证该实验的公平性和合理性, 本实验的所有算法的功用参数采用统一配置, 参数设置: 改进鲸鱼优化算法中 $k=2, d=0.5, b=1, \vec{r}$ 取值为 0 到 1 之间的随机数, \vec{a} 的取值从 2 线性递减到 0。

实验选取了 2 个种群智能算法, 分别为 PSO^[10]、FFA 以及基本鲸鱼优化算法 (WOA) 作为改进的鲸鱼优化算法的对比实验。设置所有运用到种群大小为 30, 最大能进行迭代的次数设置为 500 次, 其他算法的私有参数均按照参考文献来设置。

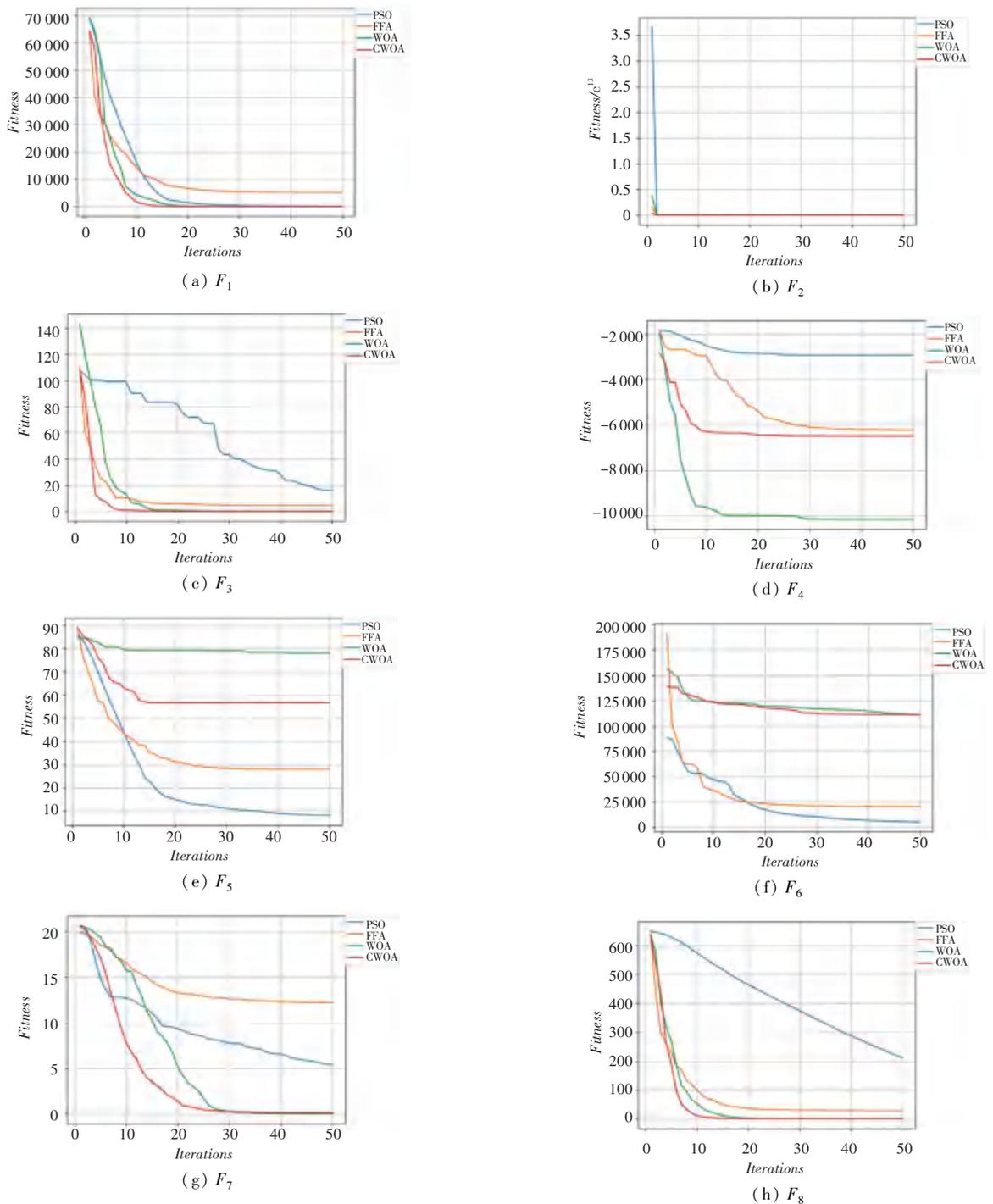
针对 8 个测试函数的性能测试实验结果见表 2。进行了对比实验, 实验结果如图 3 所示, 与 PSO、

FFA 以及基本鲸鱼优化算法 (WOA) 相比, 其均值均优于上述 3 种算法, 说明改进后的鲸鱼优化算法 (CWOA) 收敛性更好、收敛精度更高, 实验结果的稳定性更好。通过对 $F_1 \sim F_6$ 的函数变化曲线的观察, 发现在 $F_1 \sim F_3, F_7 \sim F_8$ 的函数变化曲线中, CWOA 收敛速度明显优于 PSO、FFA、WOA; 在 $F_4 \sim F_5$ 的函数变化曲线中, CWOA 的收敛速度位于比较的所有算法的中间位置。结合前文的结果可以看出, 改进后的鲸鱼优化算法 (CWOA) 的收敛精度是最高, 该算法的运行稳定性也是最好的, 收敛速度相对于其他改进的算法也比较好。

表 2 对比实验结果

Tab. 2 Comparison of experimental results

函数	PSO		FFA		WOA		CWOA	
	运行时间	最优值	运行时间	最优值	运行时间	最优值	运行时间	最优值
F_1	0.96	51.57	0.68	5 112.67	0.84	0	0.75	0
F_2	1	20.35	0.74	45.69	0.90	0	0.73	0.01
F_3	1.26	5 089.29	0.88	20 492.72	1.10	110 894	0.87	10 005.10
F_4	0.98	7.96	0	27.93	0.89	78.12	0.86	56.63
F_5	0.94	16.29	0.69	4.72	0.84	0.57	0.83	0.68
F_6	0.95	-2 933.12	0.72	-6 220.18	0.86	-10 157.10	0.80	-6 447.20
F_7	1.04	5.42	0.75	12.22	0.93	0.01	0.82	0.10
F_8	1.05	211.78	0.76	29.10	0.95	0	0.72	0

图3 $F_1 \sim F_8$ 测试函数算法性能测试图Fig. 3 $F_1 \sim F_8$ test function algorithm performance test diagram

4 结束语

鲸鱼优化算法是一种具有仿生寻优的只能优化算法,但是在对一些复杂的函数进行优化的时候,仍有一些局限性,容易陷入局部最优和收敛速度相对较慢,该改进算法中的非线性收敛因子和自适应权重策略的引进提高了算法的收敛速度,提高了鲸鱼优化算法的寻优过程,非线性收敛因子可以解决算

法在计算中全局探索能力和局部开发能力不平衡的问题,使用自适应权重策略可以使算法能够保持种群的多样性并且能够及时跳出陷入局部最优的问题,两种策略结合起来提高了算法的性能,通过在8个基准测试函数上的测试,结果表明了改进的鲸鱼优化算法能够打破陷入局部最优的限制,获得更快的收敛速度和求解精度,相比较于其他三个算法, (下转第190页)