

## 灰狼优化器

Seyedali Mirjalili<sup>a,†</sup>, Seyed Mohammad Mirjalili<sup>b</sup>安德鲁-刘易斯<sup>a</sup><sup>a</sup>澳大利亚布里斯班昆士兰州 4111 内森校园格里菲斯大学信息与通信技术学院<sup>b</sup>伊朗德黑兰沙希德·贝赫什提大学电气与计算机工程学院电气工程系, G.C. 1983963113

### Article Info

#### 文章历史:

2013 年 6 月 27 日收到

收到修订稿 2013 年 10 月 18 日接受

2013 年 12 月 11 日

2014 年 1 月 21 日在线提供

#### 关键词:

优化 优化技术 启发式

算法 元启发式算法 受

限优化 GWO

### Abstract

这项研究受灰狼 (Canis lupus) 的启发, 提出了一种名为灰狼优化算法 (GWO) 的新元启发式。GWO 算法模仿了自然界中灰狼的领导层次和狩猎机制。该算法采用了四种类型的灰狼 (如阿尔法、贝塔、德尔塔和欧米茄) 来模拟领导层次结构。此外, 还实现了狩猎的三个主要步骤: 搜索猎物、包围猎物和攻击猎物。然后在 29 个著名的测试函数上对该算法进行了基准测试, 并通过与粒子群优化 (PSO)、引力搜索算法 (GSA)、差分进化 (DE)、进化编程 (EP) 和进化策略 (ES) 的比较研究验证了结果。结果表明, 与这些著名的元启发式相比, GWO 算法能够提供非常直观的结果。本文还考虑了解决三个经典工程设计问题 (拉伸/压缩弹簧、焊接梁和压力容器设计), 并介绍了所提方法在光学工程领域的实际应用。经典工程设计问题和实际应用的结果证明, 所提出的算法适用于具有未知搜索空间的挑战性问题。

© 2013 爱思唯尔有限公司。保留所有权利。

### 1. 引言

在过去二十年里, 元启发式优化技术变得非常流行。令人惊讶的是, 其中一些技术, 如遗传算法 (GA) [1]、蚁群优化 (ACO) [2] 和粒子群优化 (PSO) [3], 不仅在计算机科学家中相

当知名, 而且在不同领域的科学家中也相当知名。除了大量的理论著作外, 这些优化技术还被应用于各个研究领域。这里有一个问题: 为什么元启发式技术变得如此普遍? 这个问题的答案可以归纳为四个主要原因: 简单性、灵活性、无派生机制和避免局部最优。

首先, 元启发式算法相当简单。它们的灵感大多来自非常简单的概念。这些灵感通常与物理现象、动物行为或进化概念有关。这种简单性允许计算机科学家模拟不同的自然概念

, 提出新的元启发式算法, 混合两种或多种元启发式算法, 或改进现有的元启发式算法。此外, 这种简单性还有助于其他科学家快速学习元启发式并将其应用于自己的问题。

其次, 灵活性是指元启发式算法可适用于不同的问题, 而无需对算法结构做任何特殊的改变。元启发式很容易适用于不同的问题。

\* 通讯作者。Tel: +61 434555738.

电子邮件地址: [seyedali.mirjalili@griffithuni.edu.au](mailto:seyedali.mirjalili@griffithuni.edu.au) (S. Mirjalili), [mohammad.smm@gmail.com](mailto:mohammad.smm@gmail.com) (S.M. Mirjalili), [a.lewis@griffith.edu.au](mailto:a.lewis@griffith.edu.au) (A. Lewis)。

因为它们大多将问题视为黑箱。换句话说，对于元启发式来说，只有系统的输入和输出是重要的。因此，设计者只需知道如何用元启发式表示他/她的问题。

第三，大多数元启发式算法都有免推导机制。与基于梯度的优化方法不同，元启发式方法是随机优化问题。优化过程从随机解开始，无需计算搜索空间的导数来找到最优解。这使得元启发式非常适用于导数信息昂贵或未知的实际问题。

最后，与传统优化技术相比，元启发式具有避免局部操作的卓越能力。这要归功于元启发式的随机性，它允许元启发

式避免局部解的停滞，并广泛搜索整个搜索空间。实际问题的搜索空间通常是未知的，而且非常复杂，存在大量局部最优，因此元启发式是优化这些具有挑战性的实际问题的良好选择。

这里值得一提的是 "无免费午餐 (NFL) " 定理<sup>[4]</sup>。该定理从逻辑上证明，不存在最适合解决所有优化问题的元启发式。换句话说，一种特定的元启发式可能在一组问题上显示出非常有前途的结果，但同样的算法在另一组问题上可能表现不佳。显然，NFL 使这一研究领域高度活跃，每年都会改进当前的方法并提出新的元启发式。这也



这促使我们从灰狼身上汲取灵感，尝试开发一种新的元启发式。

一般来说，元启发式方法可分为两大类：基于单一解决方案的和基于群体的。在前一类中（例如模拟退火法 [5]），搜索过程从一个候选解开始。然后在迭代过程中不断改进这个候选解。而基于种群的元启发式则使用一组解（种群）进行优化。在这种情况下，搜索过程从随机初始种群（多个解决方案）开始，并在迭代过程中不断改进该种群。与基于单一解决方案的算法相比，基于群体的元启发式算法具有一些优势：

- 多个候选解决方案共享搜索空间的信息，从而导致搜索空间突然跳转到有希望的部分。
- 多个候选解决方案相互协助，以避免局部最优解。
- 与基于单一解决方案的算法相比，基于群体的元启发式算法通常具有更高的利用率。

蜂群智能（SI）是基于群体的元启发式算法的一个有趣分支。SI 的概念于 1993 年首次提出[6]。根据 Bonabeau 等人的观点[1]，SI 是“简单代理群体的新兴集体智能”。SI 技术的灵感主要来源于自然界的群体、羊群、羊群和学校。一些最流行的 SI 技术包括 ACO [2]、PSO [3] 和人工蜂群 (ABC) [7]。下一节将对 SI 算法进行全面的文献综述。SI 算法的一些优点如下

- SI 算法会在迭代过程中保留搜索空间的信息，而进化算法 (EA) 则会丢弃前几代的信息。
- SI 算法通常利用内存来保存迄今为止获得的最佳解决方案。
- SI 算法需要调整的参数通常较少。
- 与进化方法（交叉、突变、精英等）相比，SI 算法的运算符较少。
- SI 算法易于实施。

无论元启发式有什么不同，其共同特点是将搜索过程分为两个阶段：探索和利用[8-12]。探索阶段指的是尽可能广泛地调查搜索空间中有希望的区域的过程。为了支持这一阶段，算法需要有随机算子对搜索空间进行随机和全局搜索。然而，利用指的是在探索阶段获得的有希望区域周围的低水平搜索能力。由于元启发式的随机性，在这两个阶段之间找到适当的平衡被认为是一项具有挑战性的任务。本文从灰狼群的社会等级和狩猎行为中汲取灵感，提出了一种新的 SI 技术。本文的其余部分安排如下：

第 2 节介绍了有关 SI 技术的文献综述。第 3 节概述了所提出的 GWO 算法。第 4-6 节分别介绍了基准函数、半现实问题

和实际应用的结果和讨论。最后，第 7 节总结了这项工作，并提出了未来研究的一些方向。

## 2. 文献综述

元启发式算法可分为三大类：进化算法、物理算法和 SI 算法。进化算法通常是

其灵感来源于自然界的进化概念。这一分支中最流行的算法是 GA

。该算法由 Holland 于 1992 年提出[13]，模拟达尔文进化概念。

Goldberg [14] 对 GA 的工程应用进行了广泛的研究。一般来说，优化是通过在 EA 中演化初始随机解来完成的。每个新种群都是由上一代个体的组合和变异产生的。由于最好的个体有更高的概率参与生成新种群，因此新种群很可能比上一代更好。这可以保证初始随机种群在世代过程中不断优化。一些 EA 包括差异进化（DE）[15]、进化编程（EP）[16,17]和进化策略（ES）[18,19]、遗传编程（GP）[20]和基于生物地理学的优化器（BBO）[21]。

例如，西蒙于 2008 年首次提出了 BBO 算法[21]。该算法的基本思想受到生物地理学的启发，生物地理学是指从地理分布（时间和空间）的角度研究生物有机体。案例研究可能包括数十年、数百年或数千年的不同岛屿、陆地甚至大陆。在这一研究领域，对不同的生态系统（栖息地或领土）进行调查，以发现不同物种（栖息地）之间在移入、移出和变异方面的关系。生态系统（考虑不同种类的物种，如捕食者和猎物）通过迁移和变异达到稳定状态的进化过程是 BBO 算法的主要灵感来源。

元启发式的第二个主要分支是基于物理的技术。这类优化算法通常模仿物理规则。其中最流行的算法有引力局部搜索（GLSA）[22]、大爆炸大压缩（BBBC）[23]、引力搜索算法（GSA）[24]、带电系统搜索（CSS）[25]、中心力优化（CFO）[26]、人工化学反应优化算法（ACROA）[27]、黑洞算法（BH）[28]、射线优化算法（RO）[29]、小世界优化算法（SWOA）[30]、星系搜索算法（GbSA）[31]和曲线空间优化算法（CSO）[32]。这些算法的机理与 EA 不同，是由一组随机的搜索代理根据物理规则在整个搜索空间中进行交流和移动。例如，这种移动是通过引力、射线投射、电磁力、惯性力、重量等来实现的。

例如，BBBC 算法的灵感来源于大爆炸和大紧缩理论。根据大爆炸理论的原理，BBBC 的搜索代理从一个点出发，在搜索空间中随机分散。它们随机搜索，然后根据大紧缩理论的原理聚集到一个最终点（迄今为止获得的最佳点）。GSA 是另一种基于物理学的算法。牛顿万有引力定律是 GSA 算法的基本物理理论依据。GSA 算法通过使用质量与适应度函数值成正比的代理集合来执行搜索。在迭代过程中，质量会因它们之间的引力而相互吸引。质量越重，吸引力越大。因此，最重的质量（可能接近全局最优值）会按照它们之间的距离吸引其他质量。

元启发式算法的第三个子类是 SI 方法。这些算法大多模仿自然界中生物群、群、群或群的社会行为。其机制几乎与基于物理的算法相似，但搜索代理利用模拟的生物集体和社会智能进行导航。最流行的 SI 技术是 PSO。PSO 算法由肯尼迪和埃伯哈特提出[3]，灵感来自鸟类成群结队的社会行为。PSO 算法采用多个粒子，这些粒子追逐最佳粒子的位置及其

迄今为止获得的最佳位置。换句话说，在移动粒子时，既要考虑粒子自身的最佳方案，也要考虑粒子群获得的最佳方案。

另一种流行的 SI 算法是 ACO，由 Dorigo 等人于 2006 年提出[2]。该算法的灵感来源于蚁群中蚂蚁的社会行为。事实上，蚂蚁在寻找巢穴和食物来源之间最短路径的社会智能是 ACO 的主要灵感来源。在迭代过程中，候选方案会演化出信息素矩阵。ABC 是另一种流行的算法，它模仿蜜蜂寻找食物源的集体行为。ABC 算法中有三种蜜蜂：侦察蜂、观察蜂和雇佣蜂。侦察蜂负责探索搜索空间，而旁观蜂和雇佣蜂则利用侦察蜂找到的有希望的解决方案。最后，最近有人提出了蝙蝠启发算法（BA），其灵感来自于蝙蝠的回声定位行为[33]。自然界中有许多种类的蝙蝠。它们的体型和重量各不相同，但在导航和捕猎时的行为却十分相似。蝙蝠利用自然声纳进行导航和捕猎。在设计 BA 算法时，我们采用了蝙蝠在寻找猎物时的两个主要特征。当蝙蝠追逐猎物时，往往会降低超声波的响度并提高发射速率。这种行为已在 BA 算法中进行了数学建模。迄今为止提出的其他 SI 技术如下：

- 2001 年蜜蜂优化算法（MBO）中的婚姻 [34]。
- 2003 年的人工鱼群算法（AFSA） [35]。
- 2005 年的白蚁算法[36]。
- 2007 年的黄蜂群算法[37]。
- 2007 年的 "猴子搜索"[38]。
- 2008 年的蜜蜂采集花粉算法（BCPA） [39]。
- 2009 年的布谷鸟搜索（CS） [40]。
- Dolphin Partner Optimization (DPO) 于 2009 年推出[41]。
- 2010 年的萤火虫算法（FA） [42]。
- 鸟类交配优化器（BMO），2012 年[43]。
- 磷虾群（KH） [44]。
- 2012 年的果蝇优化算法（FOA） [45]。

这份清单显示，迄今为止已经提出了许多 SI 技术，其中许多技术的灵感来自狩猎和搜索行为。然而，据我们所知，文献中还没有一种 SI 技术是模仿灰狼的领导层次结构的，而灰狼是以群居狩猎而闻名的。这促使我们尝试对灰狼的社会行为进行数学建模，提出一种受灰狼启发的新 SI 算法，并研究其解决基准问题和实际问题的能力。

### 3. 灰狼优化器（GWO）

本节首先讨论了拟议方法的灵感来源。然后，提供数学模型。

#### 3.1. 灵感

灰狼（*Canis lupus*）属于犬科。灰狼被认为是顶级掠食者，这意味着它们处于食物链的顶端。灰狼大多喜欢群居。狼群的平均规模为 5-12 只。特别值得注意的是，如图 1 所示，灰狼有非常严格的社会支配等级制度。

领头的是一公一母，称为首领。首领主要负责决定狩猎、睡觉地点、起床时间等。首领的决定由族群决定。不过，也有一些民主行为

据观察，首领会跟随狼群中的其他狼。在聚会时，整个狼群都会低头向首领致意。首领狼也被称为支配狼，因为狼群必须服从首领狼的命令 [46]。头狼只能在狼群中交配。有趣的是，头狼不一定是狼群中最强壮的成员，但却是管理狼群的最佳人选。这表明，狼群的组织 and 纪律远比实力重要。

灰狼等级制度的第二级是贝塔。贝塔是狼群中的从属，它们帮助首领进行决策或其他狼群活动。灰太狼可以是公的，也可以是母的，它可能是首领的最佳人选，以防首领去世或年迈。贝塔狼应该尊重首领，但也要指挥其他低级别的狼。它既是首领的顾问，也是狼群的反对者。它在整个狼群中强化首领的命令，并向首领提供反馈。

等级最低的灰狼是欧米茄。欧米茄扮演着替罪羊的角色。欧米茄狼总是要服从其他所有占统治地位的狼。它们是最后被允许进食的狼。 $\Omega$ 在狼群中似乎并不重要，但据观察，一旦失去 $\Omega$ ，整个狼群就会面临内斗和问题。这是因为欧米茄会发泄所有狼的暴力和不满。这有助于满足整个狼群的需要和维持统治结构。在某些情况下， $\Omega$ 也是狼群中的保姆。

如果一只狼不是阿尔法狼、贝塔狼或欧米茄狼，它/她就被称为次阿尔法狼（或某些参考资料中的德尔塔狼）。德尔塔狼必须服从阿尔法狼和贝塔狼，但它们主宰着欧米茄狼。童子军、哨兵、长老、猎人和看护人都属于这一类。侦察员负责监视领地的边界，并在发生危险时向狼群发出警告。哨兵负责保护狼群的安全。长老是经验丰富的狼，他们曾经是首领或副首领。猎人帮助首领和首领捕猎猎物，为狼群提供食物。最后，看护者负责照顾狼群中虚弱、生病和受伤的狼。

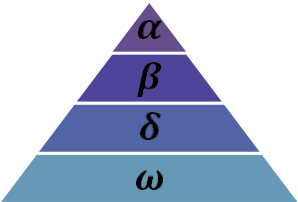
除了狼的社会等级制度外，集体狩猎也是灰狼另一种有趣的社会行为。根据 Muro 等人的研究[47]，灰狼狩猎的主要阶段如下：

- 追踪、追逐和接近猎物。
- 追逐、包围和骚扰猎物，直到猎物停止移动。
- 向猎物攻击。

这些步骤如图 2 所示。

在这项工作中，对这种狩猎技术和灰狼的社会等级制度进行了数学建模，以便设计 GWO 并进行优化。

图 1.灰狼的等级制度（优势自上而下递减）。





### 3.2. 数学模型和算法

本小节提供了社会等级、追踪、包围和攻击猎物的数学模型。然后概述 GWO 算法。

#### 3.2.1. 社会等级

在设计 GWO 时，为了对狼的社会等级进行数学建模，我们将最合适的解决方案视为  $\alpha$ 。

( $\alpha$ ) 因此，第二和第三最佳解决方案被命名为分别为 beta ( $b$ ) 和 delta ( $d$ )。其余的候选解假设为 omega ( $\omega$ )。在 GWO 算法中，**狩猎** (优化) 由  $\alpha$ 、 $b$  和  $d$  引导。

#### 3.2.2. 包围猎物

如上所述，灰狼在捕猎过程中会包围猎物。为了对包围行为进行数学建模，我们提出了以下方程：

$$\vec{D} = |\vec{C} - \vec{X}_p(t)| \cdot \vec{X}(t) \quad (3.1)$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \quad (3.2)$$

其中， $t$  表示当前迭代， $\vec{A}$  和  $\vec{C}$  是系数矢量， $\vec{X}_p$  是猎物的位置矢量， $\vec{X}$  表示位置矢量。灰狼的传播媒介。

向量  $\vec{A}$  和  $\vec{C}$  的计算方法如下：

$$\vec{A} = 2\vec{a} - r_1 \cdot \vec{a} \quad (3.3)$$

$$\vec{C} = 2 - r_2 \quad (3.4)$$

其中  $\vec{a}$  的分量在迭代过程中从 2 线性递减到 0， $r_1, r_2$  是  $[0, 1]$  中的随机向量。

为了解公式 (3.1) 和 (3.2) 的效果，图 3(a) 展示了一个二维位置矢量和更新后的位置。从图中可以看出，处于  $(X, Y)$  的灰狼更新自己的位置。

见图 3

如图 3 所示。因此，灰狼可以通过公式 (3.1) 和 (3.2) 在猎物周围的任意位置更新自己的位置。

同样的概念可以扩展到一个有  $n$  维的搜索空间，灰狼将围绕迄今为止获得的最佳解决方案在超立方体 (或超球体) 中移动。

#### 3.2.3. 狩猎

灰狼能够识别猎物的位置并包围它们。狩猎通常由首领引导。贝塔和三角狼偶尔也会参与捕猎。然而，在抽象的搜索空间中，我们并不知道最佳位置 (猎物) 在哪里。为了在数学上模拟灰狼的狩猎行为，我们假设阿尔法 (最佳候选方案)、贝塔和德尔塔对猎物的潜在位置有更好的了解。因此，我们保存迄今为止获得的前三个最佳解决方案，并要求其他搜索代理 (包括  $\omega$ ) 根据最佳搜索代理的位置更新自己的位置。为此，我们提出了以下公式。

$$\vec{X}_1 = \vec{X}_\alpha - \vec{C}_1 \cdot \vec{D}_\alpha; \vec{X}_2 = \vec{X}_\beta - \vec{C}_2 \cdot \vec{D}_\beta; \vec{X}_3 = \vec{X}_\delta - \vec{C}_3 \cdot \vec{D}_\delta \quad (3.5)$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{C}_1 \cdot \vec{D}_\alpha; \vec{X}_2 = \vec{X}_\beta - \vec{C}_2 \cdot \vec{D}_\beta; \vec{X}_3 = \vec{X}_\delta - \vec{C}_3 \cdot \vec{D}_\delta \quad (3.6)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (3.7)$$

图 4 显示了搜索代理如何在二维搜索空间中根据 alpha、beta 和 delta 更新其位置。可以看到最终位置将位于一个圆圈内的随机位置，该圆圈由搜索中 alpha、beta 和 delta 的位置定义空间。换句话说，阿尔法、贝塔和德尔塔会估计猎物的位置，其他狼会随机更新它们在猎物周围的位置。

#### 3.2.4. 攻击猎物 (利用)

如上所述，灰狼会在猎物停止移动时攻击猎物，从而完成狩猎。为了在数学上模拟接近猎物的过程，我们减小  $\vec{a}$  的值。请注意

$\vec{a}$  的波动范围也会减小  $\vec{a}$ 。换句话说， $\vec{a}$  是区间  $[2a, 2a]$  中的一个随机值，其中  $a$  从 2 到 0。当  $\vec{a}$  的随机值在  $[-1, 1]$  范围内时，搜索代理的下一个位置可以是任何位置

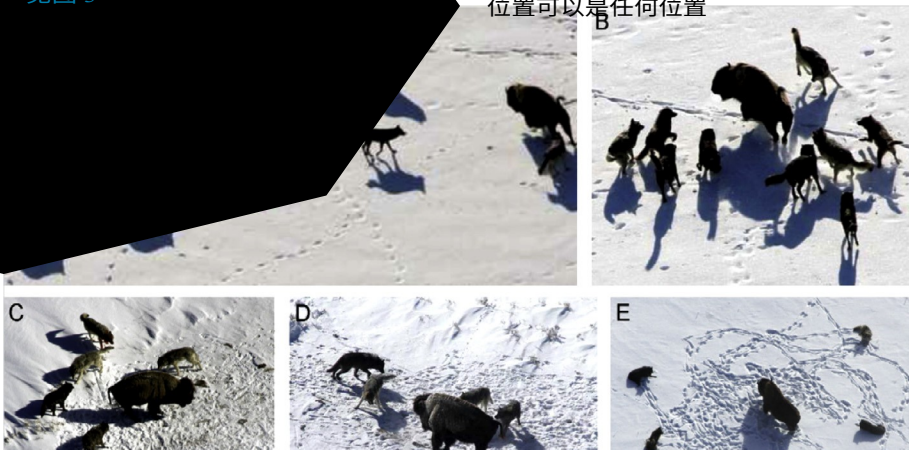




图 2 灰狼的狩猎行为：（A）追逐、接近和追踪猎物（B-D）追逐、骚扰和包围（E）静止状态和攻击 [47]。

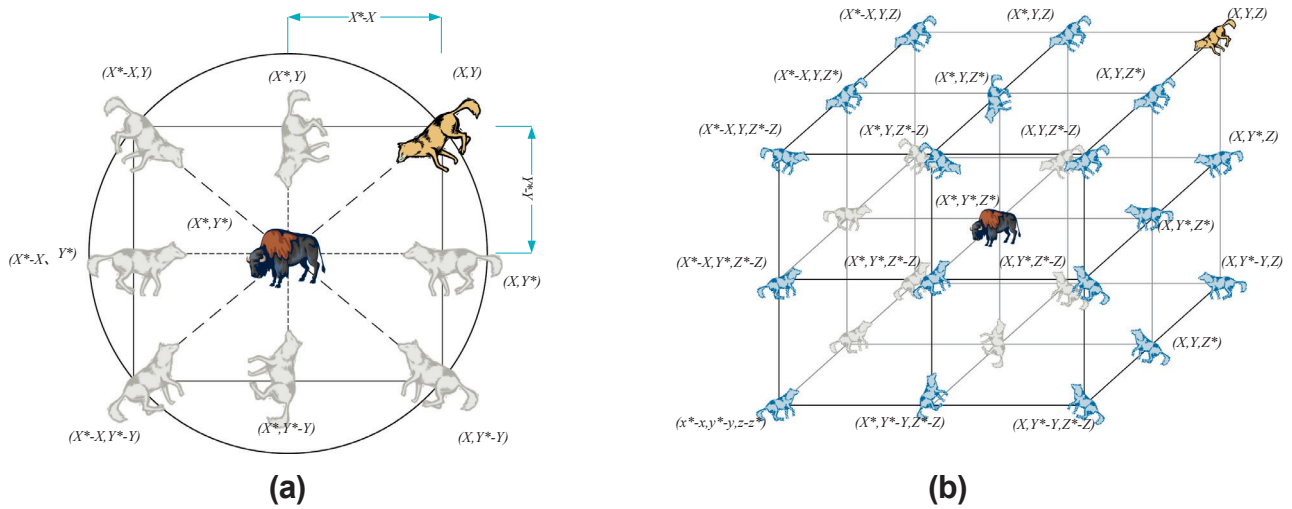


图 3：二维和三维位置矢量及其可能的下一个位置。

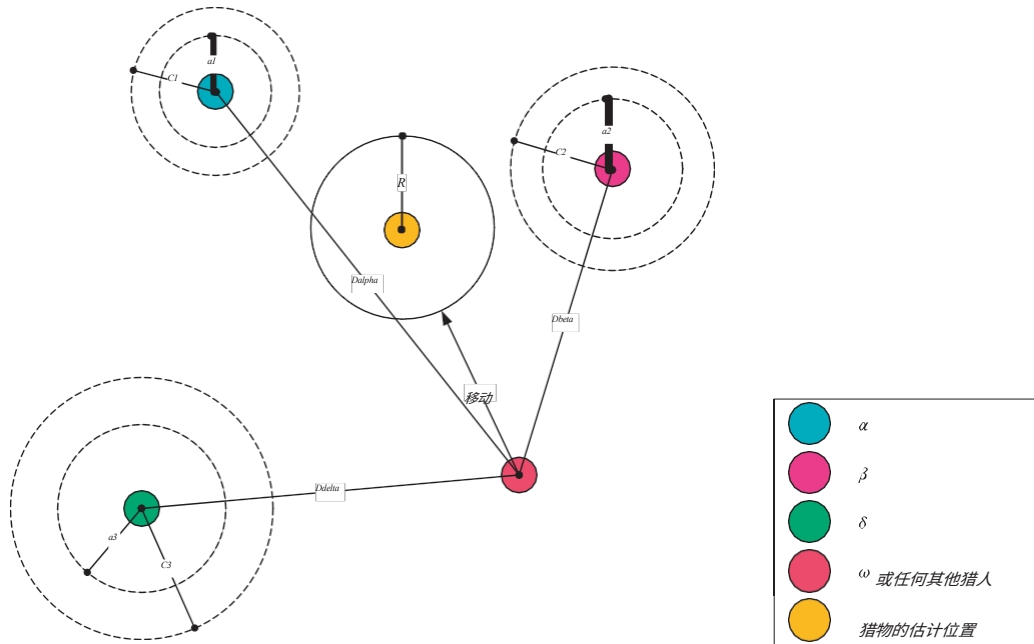


图 4.GWO 中的位置更新。

之间的距离。图 5(a) 显示， $|A| < 1$  迫使狼群向猎物攻击。

利用迄今为止提出的算子，GWO 算法允许其搜索代理根据 alpha、beta 和 delta 的位置更新其位置，并向猎物发起攻击。然而，在使用这些算子时，GWO 算法容易在局部解中停滞不前。诚然，所提出的包围机制在一定程度上体现了探索性，但 GWO 需要更多的算子来强调探索性。

### 3.2.5. 搜寻猎物（探索）

灰狼大多根据 al-pha、beta 和 delta 的位置进行搜索。它们在搜索猎物时相互发散，而在攻击猎物时相互聚合。为了在数学上模拟我们利用  $\sim A$  的随机值大于 1 或小于 -1 时，搜索代理必须偏离猎物。这强调了探索，并允许

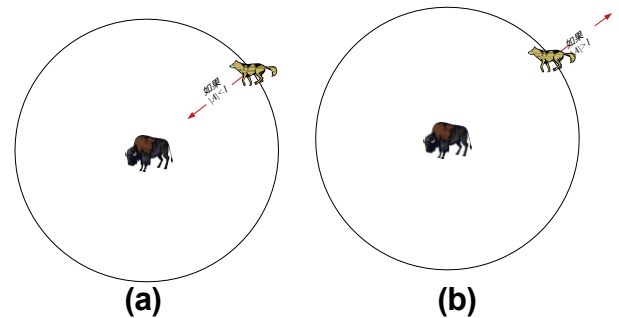


图 5.攻击猎物与搜寻猎物。

GWO 算法进行全局搜索。图 5(b) 还显示， $|A| > 1$  会迫使灰

狼偏离猎物，从而有望找到更合适的猎物。*Shirjalili et al. / Advances in Engineering Software* 全球变暖潜能值的另一个有利于勘探的成分是  $C$ 。正如从公式 (3.4) 中可以看出， $\vec{r}$  向量包含  $[0, 2]$  中的随机值。这部分内容为猎物提供随机权重，以便在捕食过程中对猎物进行统计分析。

在定义公式 (3.1) 中的距离时，可适当强调 ( $C > 1$ ) 或减弱 ( $C < 1$ ) 猎物的影响。这有助于 GWO 显示

在整个优化过程中， $C$  的行为更具随机性，有利于探索和避免局部最优。值得一提的是，与  $A$  相比， $C$  并不是线性递减的。我们特意要求  $C$  在任何时候都提供随机值，以便不仅在初始迭代中，而且在最终迭代中强调探索。在出现局部最优的情况下，尤其是在最后迭代时，这部分非常有用。

$C$  向量也可视为自然界中接近猎物的障碍物的影响。一般来说，自然界中的障碍物会出现在狼的狩猎路径上，事实上会阻碍它们快速、便捷地接近猎物。这就是矢量  $C$  的前。根据狼的位置，它可以随机给猎物加权，使狼更难更远处地接近猎物，反之亦然。

总而言之，搜索过程首先是在 GWO 算法中随机创建一个灰狼种群（候选方案）。在迭代过程中， $\alpha$ 、 $\beta$  和  $\delta$  灰狼会估计猎物的可能位置。每个候选方案都会更新与猎物的距离。参数  $a$  从 2 降为 0，以强调探索和利用、分别为候选解往往会偏离猎物。当  $|A| \geq 1$  时，GWO 算法向猎物收敛；当  $|A| < 1$  时，GWO 算法向猎物收敛。

图 6 列出了 GWO 算法的伪代码。

要了解 GWO 如何从理论上解决优化问题，有几点值得注意：

- 所提出的社会层次结构可以帮助 GWO 在迭代过程中保存迄今为止获得的最佳解决方案。
- 所提出的包围机制在解决方案周围定义了一个圆环状邻域，该邻域可以扩展到更高维度，成为一个超球体。
- 随机参数  $A$  和  $C$  可以帮助候选解具有不同随机半径的超球。
- 所提出的狩猎方法允许候选解决方案定位猎物的可能位置。
- 探索和开发由  $a$  和  $A$  的适应值来保证。
- 参数  $a$  和  $A$  的自适应值允许 GWO 在探索和开发之间平稳过渡。
- 随着  $A$  的递减，一半的迭代用于探索 ( $|A| \geq 1$ )，另一半用于开发 ( $|A| < 1$ )。
- 全球定位系统只有两个主要参数需要调整 ( $a$  和  $b$ )。

有可能整合变异和其他进化运算符来模拟灰狼的整个生命周期。

```

Initialize the grey wolf population  $X_i$  ( $i = 1, 2, \dots, n$ )
Initialize  $a$ ,  $A$ , and  $C$ 
Calculate the fitness of each search agent
 $X_a$  = the best search agent
 $X_b$  = the second best search agent
 $X_c$  = the third best search agent
while ( $t < \text{Max number of iterations}$ )
    for each search agent
        Update the position of the current search agent by equation (3.7)
    end for
    Update  $a$ ,  $A$ , and  $C$ 

```

不过，我们还是尽可能简化了 GWO 算法，减少了需要调整的运算符。建议在今后的工作中采用这种机制。该算法的源代码见 <http://www.alimirjalili.com/GWO.html> 和 <http://www.mathworks.com.au/matlabcentral/fileexchange/44974>。

#### 4. 结果和讨论

本节将在 29 个基准函数上对 GWO 算法进行基准测试。前 23 个基准函数是许多研究人员使用的经典函数 [16,48–51,82]。尽管很简单，但我们还是选择了这些测试函数，以便将我们的结果与当前的元启发式进行比较。表 1-3 列出了这些基准函数，其中  $Dim$  表示函数的维度， $Range$  是函数搜索空间的边界， $f_{min}$  是最优值。我们选择的其他测试平台是 CEC 2005 特别会议 [52] 中的六个复合基准函数。这些基准函数是经典函数的移位、旋转、扩展和组合变体，在当前的基准函数中具有最大的复杂性 [53]。表 4 列出了 CEC 2005 测试函数，其中  $Dim$  表示函数的维度， $Range$  是函数搜索空间的边界， $f_{min}$  是最优值。图 7-10 展示了所用基准函数的二维版本。

一般来说，使用的基准函数都是最小化函数，可分为四类：单模态函数、多模态函数、固定维度多模态函数和复合函数。请注意，关于复合基准函数的详细说明，请参阅 CEC 2005 技术报告 [52]。GWO 算法在每个基准函数上运行了 30 次。统计结果（平均值和标准偏差）见表 5-8。为验证结果，将 GWO 算法与基于 SI 技术的 PSO [3] 和基于物理的 GSA [24] 进行了比较。此外，GWO 算法还与三种 EA 进行了比较：DE [15]、快速进化编程（FEP）[16] 和带有协方差矩阵适应性的进化策略。

CMA-ES) [18]。

##### 4.1. 开发分析

根据表 5 的结果，GWO 能够提供极具竞争力的结果。该算法在  $F1$ 、 $F2$  和  $F7$  方面都优于其他算法。值得注意的是，单模态函数适合作为开发的基准。因此，这些结果表明，GWO 在最优开发方面表现出色。这要归功于前面讨论的开发算子。

##### 4.2. 勘探分析

与单模态函数相比，多模态函数有许多局部最优点，其数量随维度的增加呈指数增长。因此，多模态函数适合作为

表 1  
单模态基准函数

	尺寸	范围	$f_{\min}$
$f_1(x) = \sum_{i=1}^{Pn} x_i^2$	30	[-100, 100]	0
$f_2(x) = \sum_{i=1}^{Pn}  x_i  + Qn \sum_{i=1}^n  x_i $	30	[-10, 10]	0
$f_3(x) = \sum_{i=1}^{Pn} \left( \sum_{j=1, j \neq i}^n x_j \right)^2$	30	[-100, 100]	0
$f_4(x) = \max_i \{  x_i ; 1 \leq i \leq n \}$	30	[-100, 100]	0
$f_5(x) = \sum_{i=1}^{Pn-1} [100(x_{i+1} - x_i)^2 + (x_i - 1)^2]$	30	[-30, 30]	0
$f_6(x) = \sum_{i=1}^n ([x_i + 0.5])^2$	30	[-100, 100]	0
$f_7(x) = \sum_{i=1}^n  x_i ^4 + random[0; 1)$	30	[-1.28, 1.28]	0

图 6.GWO 算法的伪代码。

表 2

多模式基准功能。

功能	尺寸	范围	$f_{\min}$
$F_8(x) = \prod_{i=1}^n \sin\left(\frac{x_i}{2}\right)$	30	[-500, 500]	-418.9829 × 5
$F_9(x) = \prod_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	[-5.12, 5.12]	0
$F_{10}(x) = 20 \exp\left(-0.2 \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e\right)$	30	[-32, 32]	0
$F_{11}(x) = \frac{1}{4000} \prod_{i=1}^n x_i^2 - \prod_{i=1}^n \cos x_i + 1$	30	[-600, 600]	0
$F_{12}(x) = \prod_{i=1}^n \{10 \sin(\pi y_i) + \prod_{j=1}^{n-1} (y_j - 1)^2 [1 + 10 \sin^2(\pi y_i + 1)] + (y_n - 1)^2\} + \prod_{i=1}^n u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i - 1}{n-1}$ $u(x_i, a, k, m) = \begin{cases} 0 & 8 < k(x_i - a)^m, x_i > a \\ k(-x_i - a)^m & x_i \leq a \end{cases}$	30	[-50, 50]	0
$F_{13}(x) = 0.1 \left\{ \sin^2(3\pi x_1) + \prod_{i=1}^{n-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \right\} + \prod_{i=1}^n u(x_i, 5, 100, 4)$	30	[-50, 50]	0
$F_{14}(x) = -\prod_{i=1}^n \sin(x_i) - \sin\left(\frac{\pi}{2} \prod_{i=1}^n x_i\right), \quad x_i \in [0, 1]$	30	[0, 1]	-4.687
$F_{15}(x) = e^{\frac{1}{10} \sum_{i=1}^n x_i} - 2e^{-\frac{1}{10} \sum_{i=1}^n x_i^2} - \prod_{i=1}^n \cos^2 \frac{x_i}{2}, \quad m = 5$	30	[-20, 20]	-1
$F_{16}(x) = \left\{ \prod_{i=1}^n \sin^2(x_i) \right\} - \exp\left(-\prod_{i=1}^n x_i^2\right) - \exp\left[-\prod_{i=1}^n \sin^2\left(\frac{\pi}{2} \prod_{j=1}^n x_j\right)\right]$	30	[-10, 10]	1

表 3

固定维度多模态基准函数。

功能	尺寸	范围	$f_{\min}$
$F_{14}(x) = 1 + \frac{1}{500} \prod_{j=1}^{25} \frac{1}{\left  \frac{x_j - a}{b_j - a} \right ^2}$	2	[-65, 65]	1
$F_{15}(x) = \prod_{i=1}^{11} \left[ a_i - 1 + \frac{x_i (b_i^2 + b_i x_i)}{b_i^2 + b_i x_i + a_i} \right]$	4	[-5, 5]	0.00030
$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{5x_1^6} + x_1 x_{12} - 4x_2^2 + 4x_4^2$	2	[-5, 5]	-1.0316
$F_{17}(x) = x - \frac{5.1}{2} x^2 + \frac{5}{48} x - \frac{6}{2} \prod_{i=1}^n 10^9 1 - \cos x + \frac{1}{10}$	2	[-5, 5]	0.398
$F_{18}(x) = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_2^2 + 48x_2 - 36x_1 x_{12} + 27x_2^2)]^{\frac{2(-2, 2)}{2}}$	2	[-2, 2]	3
$F_{19}(x) = -\prod_{i=1}^4 c_i \exp(-\prod_{j=1}^3 a_{ij} (x_j - p_{ij})^2)$	3	[1, 3]	-3.86
$F_{20}(x) = -\prod_{i=1}^4 c_i \exp(-\prod_{j=1}^6 a_{ij} (x_j - p_{ij})^2)$	6	[0, 1]	-3.32
$F_{21}(x) = -\prod_{i=1}^5 [(X - a)(X - a)^T + c]^{-1}_i$	4	[0, 10]	-10.1532
$F_{22}(x) = -\prod_{i=1}^7 [(X - a)(X - a)^T + c]^{-1}_i$	4	[0, 10]	-10.4028
$F_{23}(x) = -\prod_{i=1}^{10} [(X - a)(X - a)^T + c]^{-1}_i$		[0, 10]	-10.5363

算法的探索能力。根据表 6 和表 7 的结果，GWO 在多模态基准函数上也能提供极具竞争力的结果。该算法在大多数多模态函数上的表现都优于 PSO 和 GSA。此外，与 DE 和 FEP 相比，GWO 显示出非常有竞争力的结果，偶尔还能超越它们。这些结果表明，GWO 算法在探索方面具有优势。

#### 4.3. 避免局部最小值

第四类基准函数包括复合函数，通常是元启发式算法极具挑战性的测试平台。因此，复合函数可以同时探索和利用进行基准测试。此外，由于此类测试函数中存在大量局部最优解，因此可以对算法的局部最优解避免能力进行检验。

表 8 显示，在一半的复合基准函数中，GWO 的表现优于其他所有算法。在其余的复合基准函数中，GWO 算法也提供了极具竞争力的结果。这表明，GWO 在探索和利用之间取得了良好的平衡，从而避免了大量局部最优。如上所述，一半的迭代投票用于探索 ( $|A| \geq 1$ )，其余用于开发 ( $|A| < 1$ )。

这种机制有助于 GWO 同时提供非常好的探索、避免局部极小值和开发能力。

#### 4.4. 收敛行为分析

本小节将对 GWO 的收敛行为进行研究。根据 Berg 等人的研究[54]，在优化的初始阶段，搜索代理的移动应该会有突然的变化。这有助于元启发式广泛探索搜索空间。然后，这些变化应减少，以强调优化结束时的利用。为了观察 GWO 算法的收敛行为，图 11 展示了第一个搜索代理在第一个维度的搜索历史和轨迹。该图的动画版本可在“补充材料”中找到。请注意，本节中的基准函数是移动的，我们使用了六个搜索代理来寻找最优值。

图 11 的第二列描述了搜索代理的搜索历史。可以看出，GWO 的搜索代理倾向于广泛搜索搜索空间的有希望区域，并利用最好的区域。此外，图 11 的第四列显示了第一个粒子的轨迹，其中可以观察到第一个搜索代理在其第一个维度上的变化。可以看出，在迭代的初始阶段会出现突然的变化，而随着迭代的进行，这种变化会逐渐减小。根据 Berg 等人的研究[54]，这种行为可以保证 SI 算法最终收敛到搜索空间中的某一点。

总之，与众所周知的元启发式相比，这些结果验证了 GWO 算法在求解各种基准函数时的性能。为了进一步研究



表 4

复合基准函数。

功能	尺寸	范围	$f_{\min}$
$F_{24} (CF1):$ $f_1, f_2, f_3, \dots, f_{10} =$ 球函数 $[r_1, r_2, r_3, \dots, r_{23}, \dots, r_{10}] = [1, 1, 1, \dots, 1]$ $[K_1, K_2, K_3, \dots, K_{10}] = [5/100, 5/100, 5/100, \dots, 5/100]$	10[-5, 5]	0	
$F_{25} (CF2):$ $f_1, f_2, f_3, \dots, f_{10} =$ Griewank 函数 $[r_1, r_2, r_3, \dots, r_{23}, \dots, r_{10}] = [1, 1, 1, \dots, 1]$ $[K_1, K_2, K_3, \dots, K_{10}] = [5/100, 5/100, 5/100, \dots, 5/100]$	10[-5, 5]	0	
$F_{26} (CF3):$ $f_1, f_2, f_3, \dots, f_{10} =$ Griewank 函数 $[r_1, r_2, r_3, \dots, r_{23}, \dots, r_{10}] = [1, 1, 1, \dots, 1]$ $[K_1, K_2, K_3, \dots, K_{10}] = [1, 1, 1, \dots, 1]$	10[-5, 5]	0	
$F_{27} (CF4):$ $f_1, f_2 =$ 阿克雷函数 $f_3, f_4 =$ Rastrigin 函数 $f_5, f_6$ $=$ Weierstrass 函数 $f_7, f_8 =$ Griewank 函数 $f_9, f_{10} =$ Sphere 函数 $[r_1, r_2, r_3, \dots, r_{23}, \dots, r_{10}] = [1, 1, 1, \dots, 1]$ $[K_1, K_2, K_3, \dots, K_{10}] = [5/32, 5/32, 1, 1, 5/0.5, 5/0.5, 5/100, 5/100, 5/100, 5/100]$	10[-5, 5]	0	
$F_{28} (CF5):$ $f_1, f_2 =$ Rastrigin 函数 $f_3, f_4 =$ Weierstrass 函数 $f_5, f_6$ $=$ Griewank 函数 $f_7, f_8 =$ Ackley 函数 $f_9, f_{10} =$ 球函数 $[r_1, r_2, r_3, \dots, r_{23}, \dots, r_{10}] = [1, 1, 1, \dots, 1]$ $[K_1, K_2, K_3, \dots, K_{10}] = [1/5, 1/5, 5/0.5, 5/0.5, 5/100, 5/100, 5/32, 5/32, 5/100, 5/100]$	10[-5, 5]	0	
$F_{29} (CF6):$ $f_1, f_2 =$ Rastrigin 函数 $f_3, f_4 =$ Weierstrass 函数 $f_5, f_6$ $=$ Griewank 函数 $f_7, f_8 =$ Ackley 函数 $f_9, f_{10} =$ 球函数 $[r_1, r_2, r_3, \dots, r_{23}, \dots, r_{10}] = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]$ $[K_1, K_2, K_3, \dots, K_{10}] = [0.1 * 1/5, 0.2 * 1/5, 0.3 * 5/0.5, 0.4 * 5/0.5, 0.5 * 5/100, 0.6 * 5/100, 0.7 * 5/32, 0.8 * 5/32, 0.9 * 5/100, 1 * 5/100]$	10[-5, 5]	0	

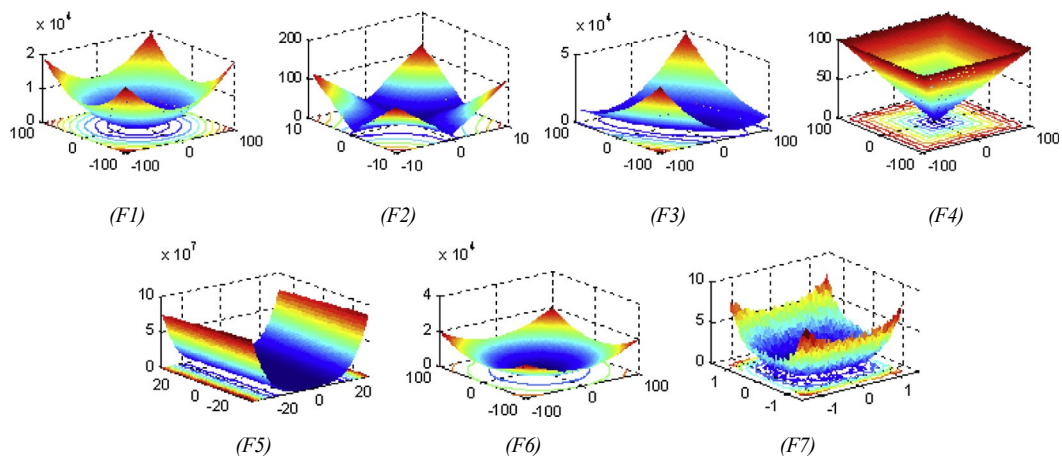


图 7. 单模态基准函数的二维版本。

在接下来的章节中，我们将利用三个经典工程设计问题和一个光学工程中的实际问题来验证所提出的算法。此外，还将 GWO 算法与知名技术进行了比较，以确认其结果。

## 5. 经典工程问题的 GWO

本节将讨论三个受限工程设计问题：拉伸/压缩弹簧、焊接梁和压力容器。

设计。这些问题都有若干相等和不相等的约束条件，因此，GWO 应配备一种约束处理方法，以便也能优化受约束的问题。一般来说，当适配函数直接影响搜索代理的位置更新时，约束处理就变得非常具有挑战性（例如 GSA）。然而，对于不依赖适度的算法来说，任何类型的约束处理都可以采用，而无需修改算法机制（例如 GA 和 PSO）。由于提议的 GWO 算法的搜索代理更新其位置是相对于

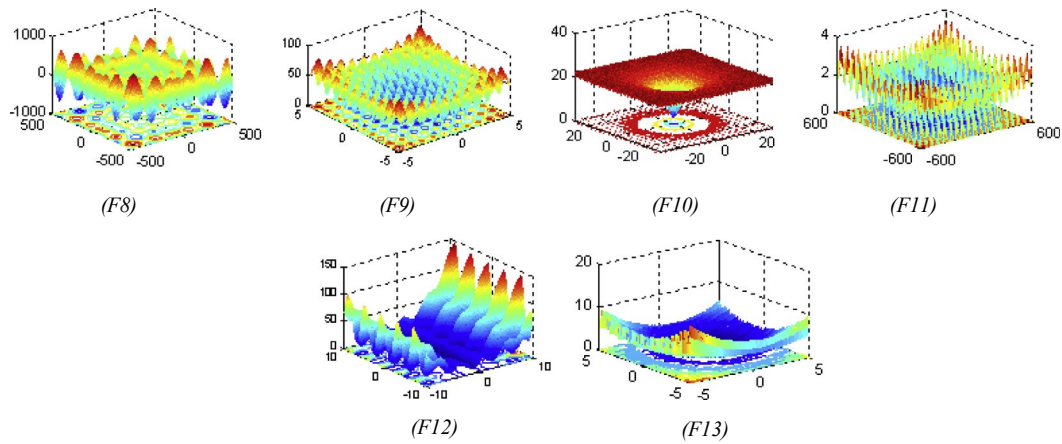


图 8. 多模态基准函数的二维版本。

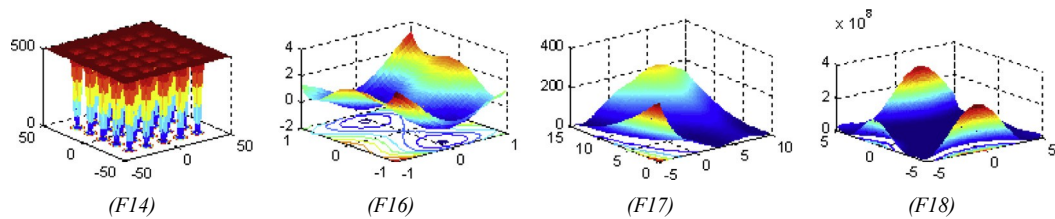


图 9. 固定维度多模态基准函数的二维版本。

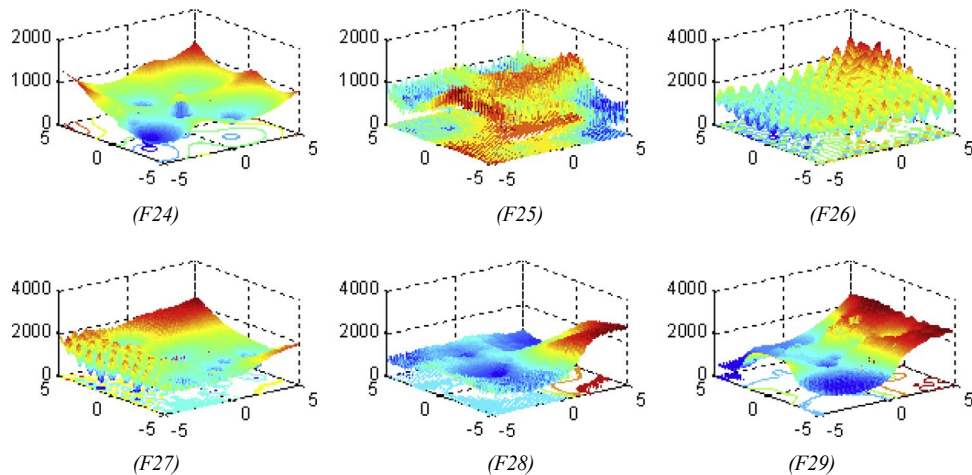


图 10. 复合基准函数的 2-D 版本。

表 5

单模态基准函数的结果。

F	GWO		PSO		GSA		DE		FEP	
	大道	标准	平均值	标准	平均值	标准	平均值	标准	平均值	标准
F1	6.59E-28	6.34E-05	0.000136	0.000202	2.53E-16	9.	67E—178.	2E—145.9E—14	0.00057	0.00013
F2	7.18E-17	0.029014	0.042144	0.045421	0.055655	0.	1940741.	5E—099.9E—10	0.0081	0.00077
F3	3.29E-06	79.14958	70.12562	22.11924	896.5347	318.	95596.	8E—117.4E—11	0.016	0.014
F4	5.61E-07	1.315088	1.086481	0.317039	7.35487	1.741452	0	0	0.3	0.5
F5	26.81258	69.90499	96.71832	60.11559	67.54309	62.22534	0	0	5.06	5.87
F6	0.816579	0.000126	0.000102	8.28E-052	.5E-16	1.74E-16	0	0	0	0
F7	0.002213	0.100286	0.122854	0.044957	0.089441	0.04339	0.00463	0.0012	0.1415	0.3522

的位置，搜索代理与适应度函数之间没有直接关系。因此，最简单的约束处理方法--惩罚函数，即搜索代理

在 GWO 中，如果 $\alpha$ 、 $\beta$ 或 $\Delta$ 违反了任何约束条件，就会被赋予较大的目标函数值，这样就可以有效地处理约束条件。在

这种情况下，如果  $\alpha$ 、 $\beta$  或  $\delta$  违反了约束条件，则它

们

表 6

多模式基准函数的结果。

F	GWO		PSO		GSA		DE		FEP	
	大道	标准	平均值	标准	平均值	标准	平均值	标准	平均值	标准
F8	-6123.1	-4087.44	-4841.29	1152.814	-2821.07	493.0375	-11080.1	574.7	-12554.5	52.6
F9	0.310521	47.35612	46.70423	11.62938	25.96841	7.470068	69.2	38.8	0.046	0.012
F10	1.06E-13	0.077835	0.276015	0.50901	0.062087	0.	236289.	7E—084.2E—08	0.018	0.0021
F11	0.004485	0.006659	0.009215	0.007724	27.70154	5.040343	0	0	0.016	0.022
F12	0.053438	0.020734	0.006917	0.026301	1.799617	0.951147	.9e -15	8e-159	.2e-063	.6e-06
F13	0.654464	0.004474	0.006675	0.008907	8.899084	7.	1262415.	1E—144.8E—140.00016		0.000073

表 7

固定维度多模态基准函数的结果。

F	GWO		PSO		GSA		DE		FEP	
	大道	标准	平均值	标准	平均值	标准	平均值	标准	平均值	标准
F14	4.042493	4.252799	3.627168	2.560828	5.859838	3.831299	0.9980043	.3E-16	1.22	0.56
F15	0.000337	0.000625	0.000577	0.000222	0.003673	0.0016474	.5E-14	0.00033	0.0005	0.00032
F16	-1.03163	-1.03163	-1.03163	6.25E-16	-1.03163	4.88E-16	-1.031633	.1E-13	-1.034	.9E-07
F17	0.397889	0.397887	0.397887	0	0.397887	0	0.3978879	.9E-09	0.3981	.5E-07
F18	3.000028	3	3	1.33E-15	3	4.17E-15	3	2E-15	3.02	0.11
F19	-3.86263	-3.86278	-3.86278	2.58E-15	-3.86278	2.29E-15	不适用	不适用		0.000014
F20	-3.28654	-3.25056	-3.26634	0.060516	-3.31778	0.023081	不适用	不适用		0.059
F21	-10.1514	-9.14015	-6.8651	3.019644	-5.95512	3.737079	-10.1532	0.0000025	-5.52	1.59
F22	-10.4015	-8.58441	-8.45653	3.087094	-9.68447	2.014088	-10.40293	.9E-07	-5.53	2.12
F23	-10.5343	-8.55899	-9.95291	1.782786	-10.53642	.6E-15	-10.53641	.9E-07	-6.57	3.14

会在下一次迭代中被新的搜索代理自动替换。为了根据搜索代理的违规程度对其进行惩罚，可以随时使用任何一种惩罚函数。在这种情况下，如果惩罚使得 alpha、beta 或 delta 的拟合度低于任何其他狼群，那么在下次迭代中，它就会被新的搜索代理自动替换。除了拉伸/压缩弹簧设计问题使用了更复杂的惩罚函数外，其余问题我们都使用了简单的标量惩罚函数。

### 5.1. 拉伸/压缩弹簧设计

如图 12 所示，该问题的目标是最小化拉伸/压缩弹簧的重量[55–57]。最小化过程受到一些约束条件的限制，如剪应力、冲击频率和最小变形。这个问题有三个变量：线径 ( $d$ )、平均线圈直径 ( $D$ ) 和有效线圈数 ( $N$ )。该问题的数学公式如下

考虑  $\rightarrow x = [x_1 \ x_2 \ x_3] = [dDN]$ 、

最小化  $f(\rightarrow x) = (x_3 + 2)x_2^2 x_1^2$ 、  
受限于  $g_1(\rightarrow x) = 1 - \frac{x_2^2}{x_3^3} \geq 0$ 、

$$g_2(\rightarrow x) = \frac{4x_2^2 - x_1 x_2}{12566(x_2^3 - x_1^4)} + \frac{1}{5108x_2} \geq 0,$$

$$g_3(\rightarrow x) = \frac{4x_2}{2} - \frac{x_1 x_2^2}{12566(x_2^3 - x_1^4)} + \frac{1}{5108x_2} \geq 0, \quad (1)$$

$$g_4(\rightarrow x) = 1 - \frac{140.423x_1}{x_2^3} \geq 0,$$

$$g_5(\rightarrow x) = \frac{x_1 + x_2}{1.5} - 1 \geq 0$$

解决这一问题所采用的数学方法有数值优化技术（在成本不变的情况下进行约束修正）[55] 和数学优化技术 [56]。这些技术与 GWO 的结果比较见表 9。请注意，为了进行公平比较，我们对 GWO 使用了类似的惩罚函数 [63]。表 9 显示，GWO 为该问题找到了权重最小的设计方案。

### 5.2. 焊接梁设计

如图 13 所示，该问题的目标是最小化焊接梁的制造成本 [60]。约束条件如下

- 剪应力 ( $S$ )。
- 梁的弯曲应力 ( $h$ )。
- 钢筋上的屈曲载荷 ( $P_c$ )。
- 梁的末端挠度 ( $d$ )。
- 侧面限制。

这个问题有四个变量，如焊缝厚度 ( $h$ )、棒材附着部分的长度 ( $l$ )、棒材的高度 ( $t$ ) 和厚度  $-----$ 。

条形图 (b)。数学公式如下

考虑  $\rightarrow x = [x_1 \ x_2 \ x_3 \ x_4] = [hltb]$ 、  
最小化  $f(\rightarrow x) = 1.10471x_1^2 x_2 + 0.04811x_3 x_4 (14.0 + x_2)$

取决于  $g_1(\rightarrow x) = S(\rightarrow x) - \max 60$ 、

$$g_2(\rightarrow x) = I(\rightarrow x) - \max 60,$$

变量范围  $0.05 \leq x_1 \leq 2.00$

$$0.25 \leq x_2 \leq$$

$$1.30、$$

$$g_4(\vec{x}) = x_1 - x_4 \in [0, 1], g_5(\vec{x}) = P - P_c(\vec{x}) \in [0, 1]$$

$$g_6(\vec{x}) = 0.125 - x_1 \in [0, 1]$$

$$2.00 \leq x_3 \leq 15.0$$

$$g_7(\vec{x}) = 1.10471x^2 + 0.04811x_3(14.0 + x_2) - 5.0 \in [0, 1]$$

数学和启发式方法都曾解决过这一问题。Ha 和 Wang 尝试用 PSO 解决这一问题 [58]。进化策略 (ES) [59]、GA[60]、和谐搜索 (HS) [61]和差分进化 (DE) [62]算法也被用作该问题的启发式优化器。这些算法

可变范围  $0.1 \leq x_1 \leq 2$ 、

$$0.1 \leq x_2 \leq 10、$$

$$0.1 \leq x_3 \leq 10、$$

$$0.1 \leq x_4 \leq 2$$

表 8

综合基准函数的结果。

F	GWO		PSO		GSA		DE		CMA-ES	
	大道	标准	平均值	标准	平均值	标准	平均值	标准	平均值	标准
F24	43.83544	69.86146	100	81.65	6.63E-17	2.78E-17	6.75E-02	1.11E-01	100	188.56
F25	91.80086	95.5518	155.91	13.176	200.6202	67.72087	28.759	8.6277	161.99	151
F26	61.43776	68.68816	172.03	32.769	180	91.89366	144.41	19.401	214.06	74.181
F27	123.1235	163.9937	314.3	20.066	170	82.32726	324.86	14.784	616.4	671.92
F28	102.1429	81.25536	83.45	101.11	200	47.14045	10.789	2.604	358.3	168.26
F29	43.14261	84.48573	861.42	125.81	142.0906	88.87141	490.94	39.461	900.26	8.32E-02

其中,  $S(-x) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \dots \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ ,

$$S' = \frac{P}{S} S^{MR}, \quad \text{其中 } L = \frac{1}{2} \left( \frac{1}{2} + \frac{1}{2} \right)$$

$$J = \frac{2n}{\pi} \int_0^{\frac{\pi}{2}} \frac{f(x)}{x} dx, \quad \text{其中 } f(x) = \frac{1}{x} \left( \frac{1}{2} + \frac{1}{2} \right)$$

$$P(k) = \frac{4.013E \times 10^6}{12} \left( \frac{1}{2} + \frac{1}{2} \right)$$

$P = 6000$  磅,  $L = 14$  英寸,  $d_{max} = 0.25$  英寸,  $E = 30 \times 10^6$  psi,  $G = 12 \times 10^6$  psi,  
 $s_{max} = 13600$  psi,  $r_{max} = 30000$  psi

Coello [64] 和 Deb [65,66] 采用了 GA, 而 Lee 和 Geem

[67] 使用 HS 解决了这一问题。Richardson 的随机法、Simplex

法、Davidon-Fletcher-Powell、Griffith 和 Stewart 的连续性线性近

似法 [68] 针对该问题采用的数学方法。

见表 10 比较结果

设计。

如图 14 所示。容器的两端都有盖, 头部呈半球形。这个问题有四个变量:

- 外壳厚度 ( $T_s$ )。
- 封头厚度 ( $T_h$ )。
- 不考虑封头的圆柱形截面长度 ( $L$ )。

这个问题有四个约束条件。这些约束条件和问题的表述如下:

考虑  $x = [x_1 \ x_2 \ x_3 \ x_4] = [T_s \ T_h \ L]$ ,最小化  $f(x) = 0.6224x_1 x_2 x_3 + 1.7781x_1 x_2^2 + 3.1661x_1 x_2^4 + 19.84x_1 x_2^3$ ,根据  $g_1(x) = -x_1 + 0.0193x_3 \leq 0$ ,

$$g_2(x) = -x_3 + 0.00954x_3 \leq 0$$

$$g_3(x) = -p x_2^2 - \frac{4}{3} p x_3^3 + 1296000 \leq 0,$$

$$g_4(x) = x_4 - 240 \leq 0,$$

(5.3)

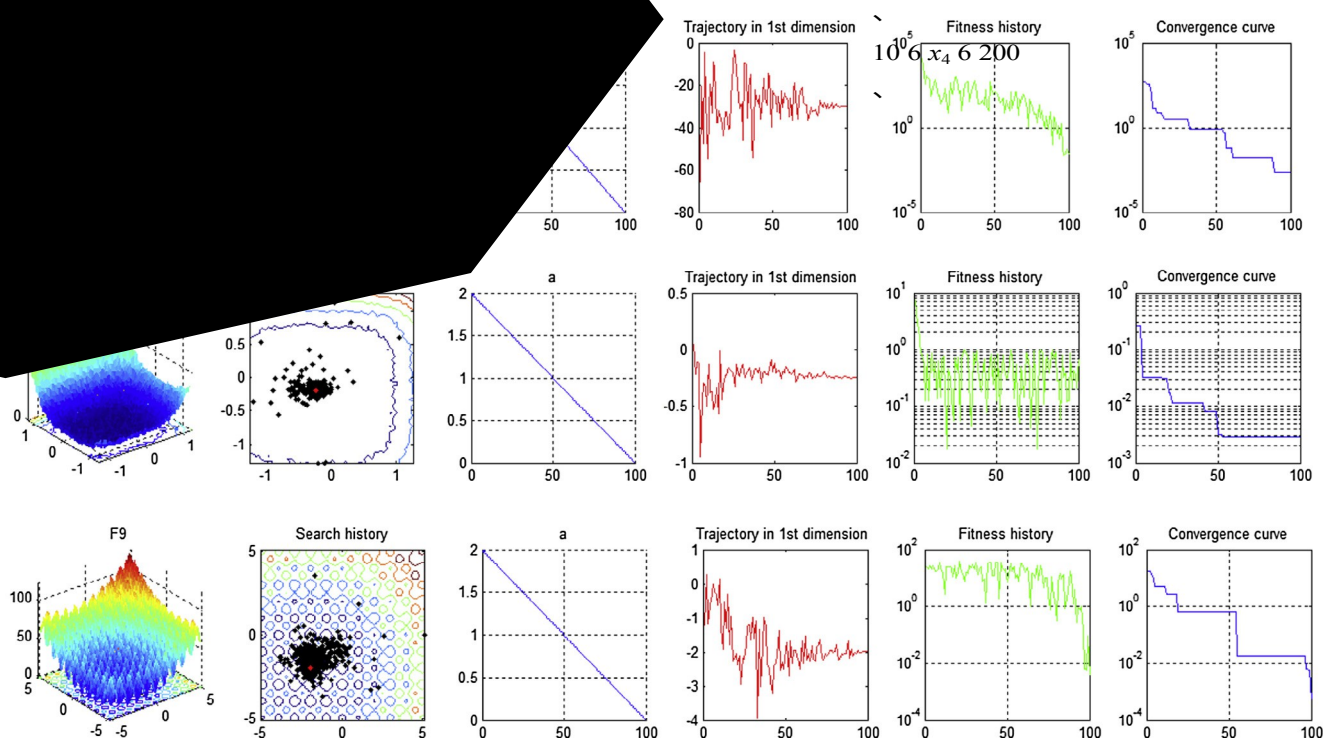
可变范围  $0 \leq x_1 \leq 99$ , $0 \leq x_2 \leq 99$ , $10 \leq x_3 \leq 200$ 



图 11. 第一个粒子在第一维度的搜索历史和轨迹。

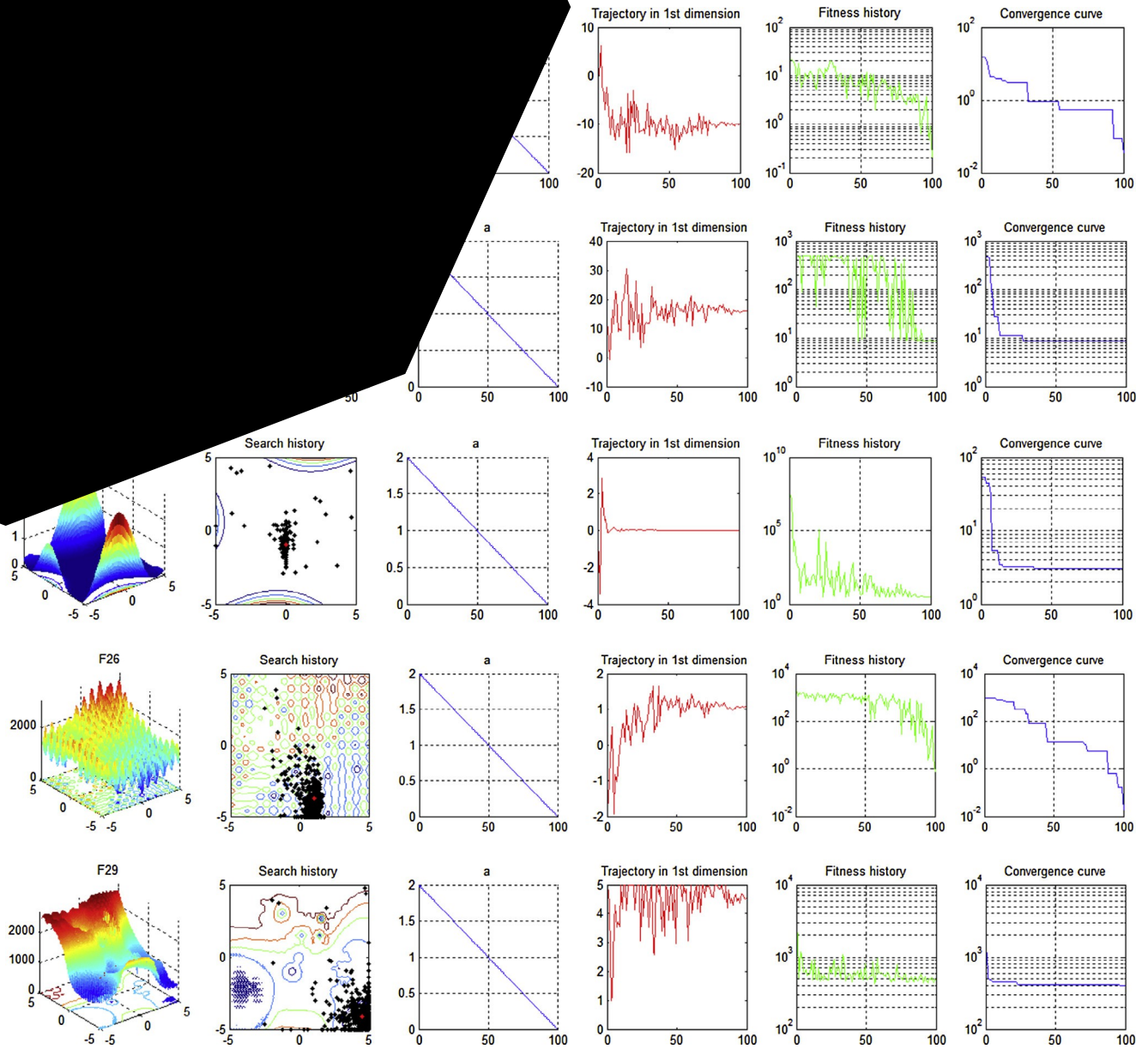


图 11 (续)

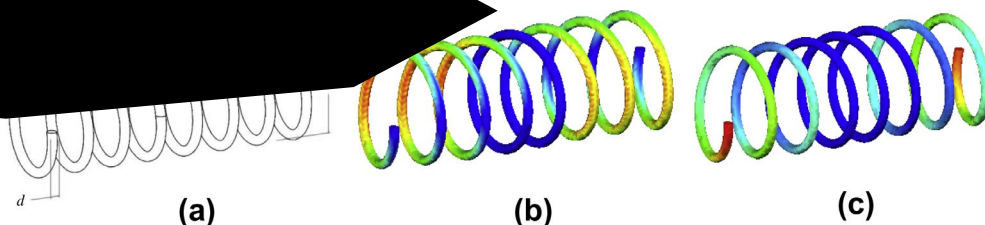


图 12. 拉伸/压缩弹簧: (a) 剪切图; (b) 应力热图; (c) 位移热图。

这一问题也受到研究人员的青睐,并在各种研究中得到优化。采用启发式方法来优化这一问题的有: PSO [58]、GA [57,60,69]、ES [59]、DE [62] 和 ACO [70]: PSO [58]、GA [57,60,69]、ES [59]、DE [62] 和 ACO [70]。使用的数学方法有增强拉格朗日乘法器 [71] 和分支-约束 [72]。该问题的结果

见表 11。根据该表, GWO 再次能够找到成本最低的设计。

8  
总之，对三个经典工程问题的研究结果表明，GWO 在解决真

*S. M. Mirjalili et al. / Advances in Engineering Software 69 (2014) 46–61*

有挑战性的问题时表现出很高的性能。这同样要归功于运算符的设计，这些运算符使 GWO 能够成功避开局部最优，并迅速达到最优。下一节将探讨 GWO 算法在解决光学工程领域最新实际问题时的性能。



表 9  
拉伸/压缩弹簧设计问题的结果比较。

算法	最佳变量			最优重量
	<i>d</i>	<i>D</i>	<i>N</i>	
全球战略组织	0.05169	0.356737	11.28885	0.012666
GSA	0.050276	0.323680	13.525410	0.0127022
PSO (Ha 和 Wang)	0.051728	0.357644	11.244543	0.0126747
ES (科埃洛和蒙特斯)	0.051989	0.363965	10.890522	0.0126810
GA (Coello)	0.051480	0.351661	11.632201	0.0127048
HS (Mahdavi et al.)	0.051154	0.349871	12.076432	0.0126706
DE (Huang et al.)	0.051609	0.354714	11.410831	0.0126702
数学优化 (贝勒古杜)	0.053396	0.399180	9.1854000	0.0127303
约束校正 (阿罗拉)	0.050000	0.315900	14.250000	0.0128334

6. GWO 在光学工程中的实际应用（光缓冲器设计）

本节研究的问题称为光缓冲器设计。事实上，光缓冲器是光 CPU 的主要组件之一。光缓冲器减缓光的群速度，使光 CPU 可以处理光包或调整其定时。最常用的设备是光子晶体波导（PCW）。PCW 大多具有晶格状结构，中间有线状缺陷。孔的半径和线缺陷的形状会产生不同的慢光特性。不同的半径和线缺陷为波导中的光线折射提供了不同的环境。该领域的研究人员试图操纵孔的半径和线缺陷的针脚，以获得理想的光缓冲特性。此外，还有不同类型的 PCW 适用于特定应用。本节将采用 GWO 算法优化一种名为布拉格槽 PCW（BSPCW）的 PCW 结构。这个问题有多个约束条件，因此我们在本节中也采用了 GWO 最简单的约束条件处理方法。

BSPCW 结构由 Caer 等人于 2011 年首次提出 [73]。BSPCW 的结构如图 15 所示。背景板是折射率等于 3.48 的硅。槽和孔由折射率为 1.6 的材料填充。布拉格槽结构使 BSPCW 能够精确控制色散和慢光特性。如文献 [73] 所述，与槽相邻的前五个孔对慢光特性的影响最大。如图 15 所示，*l*、*w<sub>l</sub>* 和 *w<sub>h</sub>* 定义了槽的几何参数，它们对慢光特性产生影响。因此，各

表 10  
焊接梁设计问题的比较结果。

算法	最佳变量				最优费用
	<i>h</i>	<i>l</i>	<i>t</i>	<i>b</i>	
全球战略组织	0.205676	3.478377	9.03681	0.205778	1.72624
GSA	0.182129	3.856979	10.00000	0.202376	1.879952
GA (Coello)	不适用	不适用	不适用	不适用	1.8245
GA (Deb)	不适用	不适用	不适用	不适用	2.3800
GA (Deb)	0.2489	6.1730	8.1789	0.2533	2.4331
HS (李和 Geem)	0.2442	6.2231	8.2915	0.2443	2.3807
随机简单	0.4575	4.7313	5.0853	0.6600	4.1185
戴维	0.2792	5.6256	7.7512	0.2796	2.5307
大约	0.2434	6.2552	8.2915	0.2444	2.3841
大约	0.2444	6.2189	8.2915	0.2444	2.3815

有两种指标可用于比较慢光设备的性能：延迟-带宽乘积（DBP）和归一化 DBP（NDBP）的定义如下 [74]：

DBP = *Dt* - *Df* (6.1)

其中，*Dt* 表示延迟，*Df* 是慢光装置的带宽。

在慢光设备中，最终目标是以最高的 PCW 带宽实现光脉冲的最大传输延迟。显然，为了提高 DBP，应增加 *Dt*。这可以通过增加器件的长度（*L*）来实现。要比较不同长度和工作频率的器件，NDBP 是更好的选择 [75]：

NDBP =  $\frac{1}{n_g} - \frac{Dx}{x_0}$  (6.2)

其中，*n<sub>g</sub>* 是群指数的平均值，*Dx* 是归一化带宽，*x<sub>0</sub>* 是光波的归一化中心频率。

由于 NDBP 与分组指数（*n<sub>g</sub>*）有直接关系，因此可表述如下 [76]：

$C_g = \frac{C}{v_g} = \frac{dkn}{63 dx}$  (6.3)

其中，*x* 表示色散，*k* 表示波矢量，*C* 表示自由空间中的光速，表示群指数。由于 *n<sub>g</sub>* 在带宽范围内不断变化，因此应按如下方法进行平均：

$n_g = \frac{1}{x_L - x_H} \int_{x_L}^{x_H} n_g(x) \frac{dx}{Dx}$  (6.4)

PCW 的带宽是指 *n<sub>g</sub>* 曲线的区域，其中 *n<sub>g</sub>* 有一个近似恒定的值，其波动最大值为 0.5%。

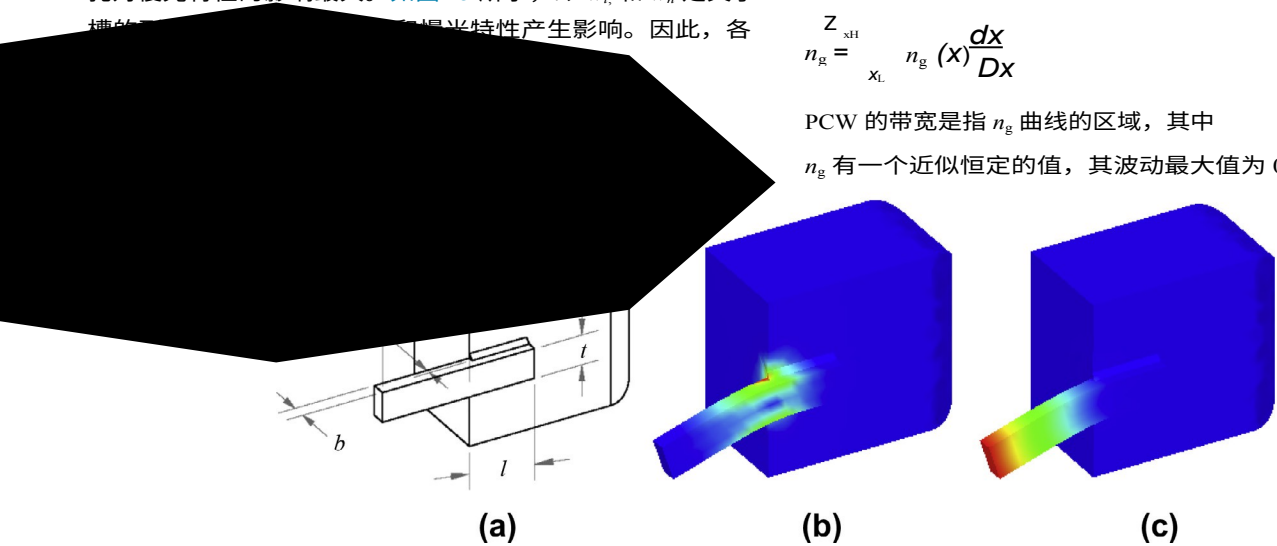


图 13.焊接梁的设计结构 (a) 剪切图 (b) 应力热图 (c) 位移热图。

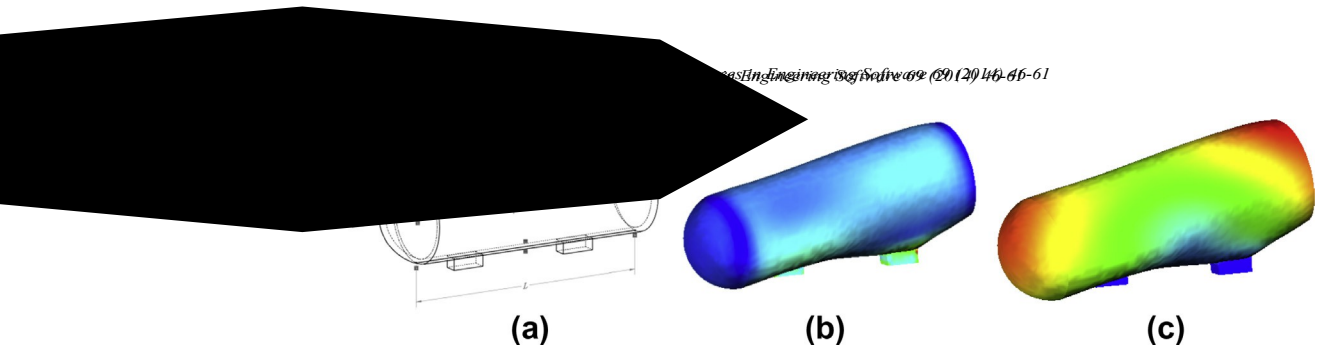


图 14.压力容器 (a) 剪切图 (b) 应力热图 (c) 位移热图。

表 11  
压力容器设计问题的比较结果。

算法	最佳变量				最佳成本
	$T_s$	$T_h$	$R$	$L$	
全球战略组织	0.812500	0.434500	42.089181	176.758731	6051.5639
GSA	1.125000	0.625000	55.9886598	84.4542025	8538.8359
PSO (He 和 Wang)	0.812500	0.437500	42.091266	176.746500	6061.0777
GA (Coello)	0.812500	0.434500	40.323900	200.000000	6288.7445
GA (Coello and Montes)	0.812500	0.437500	42.097398	176.654050	6059.9463
佐治亚州 (黛博和吉恩)	0.937500	0.500000	48.329000	112.679000	6410.3811
ES (Montes and Coello)	0.812500	0.437500	42.098087	176.640518	6059.7456
DE (Huang et al.)	0.812500	0.437500	42.098411	176.637690	6059.7340
ACO (Kaveh 和 Talataheri)	0.812500	0.437500	42.103624	176.572656	6059.0888
拉格朗日乘法器 (卡南)	1.125000	0.625000	58.291000	43.6900000	7198.0428
分支方向 (桑德格伦)	1.125000	0.625000	47.700000	117.701000	8129.1036

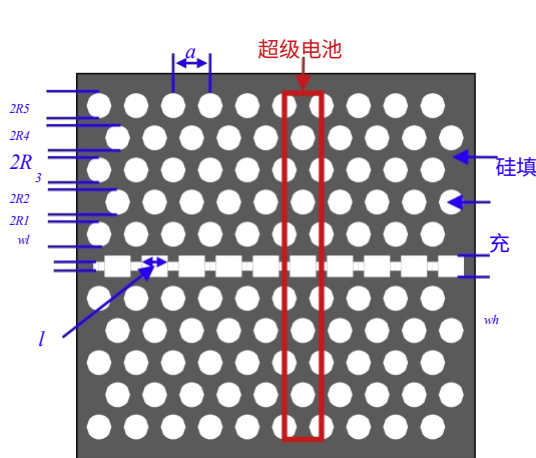


图 15.具有超级单元的 BSCPW 结构， $n_{background} = 3.48$ ， $n_{filled} = 1.6$ 。

$\pm 10\%$ [75]。有关 PCW 的详细信息，请参见文献 [77-80]。

最后，对 GWO 问题的数学表述如下：

表 12  
结构参数和计算结果

结构参数	Wu 等人[81]	全球战略组织
$R_1$	-	0.33235a
$R_2$	-	0.24952a
$R_3$	-	0.26837a
$R_4$	-	0.29498a
$R_5$	-	0.34992a
$l$	-	0.7437a
$W_h$	-	0.2014a
$W_l$	-	0.60073a
$a(\text{nm})$	430	343
$n_g$	23	19.6
$Dk(\text{nm})$	17.6	33.9

考虑： $\rightarrow x = [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 \ x_8] = \begin{bmatrix} R_1 & R_2 & R_3 & R_4 & R_5 & l & W_h & W_l \\ a & a & a & a & a & a & a & a \end{bmatrix}$ 、

最大化： $f(\rightarrow x) = \text{NDBP} = \frac{g \cdot D_x}{x_0}$ 、

条件是： $\max(|b_2(X)|) < 10^{6 \cdot a/2pc^2}$ 、

$x_H < \min(x_{up \text{ band}})$ 、

$x_L > \max(x_{down \text{ band}})$ 、

$kn > knH \rightarrow x_{\text{Guided 模式}} > x_H$ 、

$b$ 的数量级 $_2 (a/2pc)^2$	$10^3$	$10^3$
NDBP	0.26	0.43



$$k_n < k_{nL} \rightarrow X \text{ 导向模式 } < k_L \text{ (6.5)}$$

其中： $X_H = X(k_{nH}) = X(1.1n_{g0})$ 、  
 $X_L = X(k_{nL}) = X(0.9n_{g0})$ 、  
 $k_n = \frac{ka}{a}$   
 $Dx = X_H - X_L$ 、  
 $a = x_0 * 1550$ （纳米）、

2p

可变范围： $0 \leq x_1 \leq 0.5$ 、  
 $0 \leq x_6 \leq 1$ 、  
 $0 \leq x_{7,8} \leq 1$ 、

请注意，我们为 GWO 算法考虑了五个约束条件。第二至第五个约束条件避免了带状混合。为了处理可行性问题，我们给违反约束条件的搜索代理分配了较小的负目标函数值（100）。

GWO 算法在此问题上运行了 20 次，获得的最佳结果见表 12。请注意，该算法是在格里菲斯大学的 Windows HPC 集群上由 24 个 CPU 运行的。该表显示，利用 GWO 算法，带宽 (Dk) 和 NDBP 分别大幅提高了 93% 和 65%。

BSPCW 优化后的光子带结构如图 16（a）所示。此外，图 16（b）和图 17 还显示了相应的群指数和优化后的超级单元。这些数字表明，优化后的结构具有非常好的带宽，而且不存在带混合现象。这再次证明了 GWO 算法在解决实际问题时的高性能。

这项综合研究表明，拟议的 GWO 算法在当前的元启发式算法中具有优势。首先，重新

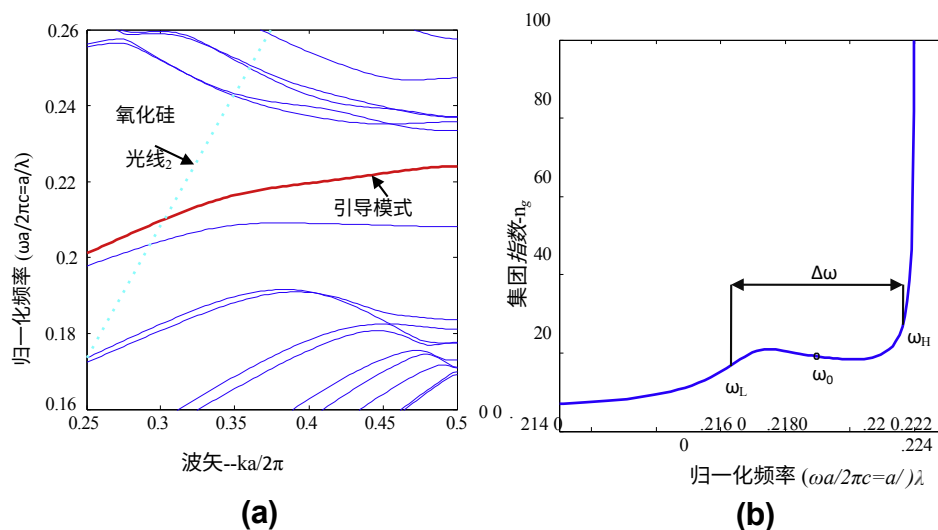


图 16.(a) 优化 BSPCW 结构的光子带结构 (b) 优化 BSPCW 结构的群指数 ( $n_g$ )。

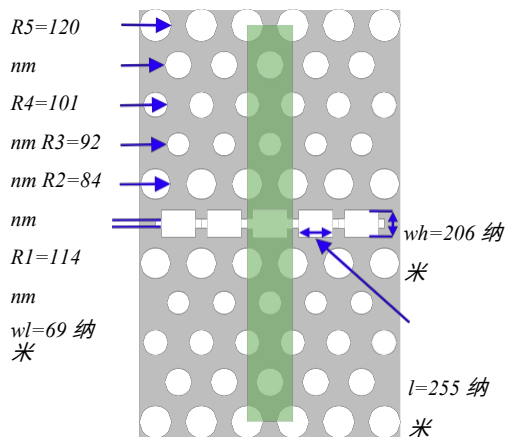


图 17.优化后的 BSPCW 超级电池。

无约束基准函数的结果证明了 GWO 算法在探索、利用、避免局部最优和收敛方面的性能。其次，经典工程问题的结果表明，所提出的算法在解决半真实的有约束问题时性能优越。最后，光缓冲器设计问题的结果显示了 GWO 算法解决实际问题的能力。

## 7. 结论

这项研究提出了一种受灰狼启发的新型 SI 优化算法。该方法模仿了灰狼的社会等级和狩猎行为。研究采用了 29 个测试函数，从探索、利用、避免局部最优和收敛等方面对所提算法的性能进行了基准测试。结果表明，与 PSO、GSA、DE、EP 和 ES 等知名启发式算法相比，GWO 能够提供极具竞

最后，GWO 的收敛性分析证实了该算法的收敛性。

此外，工程设计问题的结果还表明，GWO 算法在未知的、具有挑战性的搜索空间中具有很高的性能。GWO 算法最终被应用于光学工程中的一个实际问题。该问题的结果表明，与现有方法相比，NDBP 有了很大改进，这表明所提出的算法适用于解决实际问题。值得注意的是，在半真实和真实问题上的结果也证明，GWO 不仅在无约束问题上，而且在有约束问题上都能表现出很高的性能。

在未来的工作中，我们将开发二元和多目标版本的 GWO 算法。

争力的结果。首先，单模态函数的结果表明，GWO 算法的利用率更高。其次，多模态函数的结果证实了 GWO 的探索能力。第三，复合函数的结果显示了较高的局部最优回避能力。Fi-

## 附录 A.补充材料

与本文相关的补充数据见网络版：  
<http://dx.doi.org/10.1016/j.advengsoft.2013.12.007>.

## 参考资料

- [1] Bonabeau E, Dorigo M, Theraulaz G. Swarm intelligence: from natural to artificial systems: OUP USA; 1999.
- [2] Dorigo M, Birattari M, Stutzle T. Ant colony optimization.Comput Intell Magaz, IEEE 2006; 1: 28-39.
- [3] Kennedy J, Eberhart R. 粒子群优化, 《神经网络》, 1995 年。In: Proceedings, IEEE international conference on; 1995. p. 1942-1948.
- [4] Wolpert DH, Macready WG.没有免费午餐的优化定理.Evolut Comput, IEEE Trans 1997;1:67-82.
- [5] Kirkpatrick S, Jr.DG, Vecchi MP.通过模拟退火进行优化。Science, vol. 220; 1983. p. 671-80.
- [6] Beni G, Wang J. 蜂窝机器人系统中的群体智能。In: Robots and biological systems: towards a new bionics?, ed.. Springer; 1993.Springer; 1993. p. 703-12.
- [7] Basturk B, Karaboga D. 用于数值函数优化的人工蜂群 (ABC) 算法。In: IEEE swarm intelligence symposium; 2006. p. 12-4.
- [8] Olorunda O, Engelbrecht AP.利用粒子群多样性衡量粒子群的探索/开发。In: 进化计算, 2008 年。CEC 2008 (IEEE 计算智能世界大会)。IEEE Congress on; 2008. p.1128-34.
- [9] Alba E, Dorronsoro B. 动态细胞遗传算法中的探索/开发权衡。Evolut Comput, IEEE Trans 2005;9:126-42.
- [10] Lin L, Gen M. 进化算法的自动调整策略: 探索与开发之间的平衡。Soft Comput 2009;13:157-68.
- [11] Mirjalili S, Hashim SZM.用于函数优化的新型混合 PSOGSA 算法。In: p. 374-77.

- [12] Mirjalili S, Mohd Hashim SZ, Moradian Sardroudi H. 使用混合粒子群优化和引力搜索算法训练前馈神经网络. *Appl Math Comput* 2012;218:11125–37.
- [13] Holland JH. 遗传算法. *Sci Am* 1992;267:66–72.
- [14] Goldberg D. Genetic Algorithms in optimization, search and machine learning, Addison Wesley, New York. In: Eiben AE, Smith JE, editors. 2003 进化计算导论。Springer. Jacq J, Roux C (1995) Registration of non-segmented images using a genetic algorithm. 计算机科学讲义, 第 905 卷; 1989 年。第 205–11 页。
- [15] Storn R, Price K. 微分进化论--用于连续空间全局优化的简单高效启发式. *J Global Optim* 1997;11:341–59.
- [16] Yao X, Liu Y, Lin G. Evolutionary programming made faster. *Evolut Comput, IEEE Trans* 1999;3:82–102.
- [17] Fogel D. 《通过模拟进化的人工智能》。Wiley-IEEE Press; 2009.
- [18] Hansen N, Müller SD, Koumoutsakos P. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolut Comput* 2003;11:1–18.
- [19] Rechenberg I. Evolution strategy. *Comput Intel Imitat Life* 1994;1.
- [20] Koza JR. 遗传编程; 1992 年。
- [21] Simon D. 基于生物地理学的优化. *Evolut Comput IEEE Trans* 2008;12:702–13.
- [22] Webster B, Bernhard PJ. 基于重力自然原理的局部搜索优化算法. In: 2003 年信息与知识工程国际会议 (IKE'03) 论文集, 美国内华达州拉斯维加斯; 2003 年, 第 255–61 页。
- [23] Erol OK, Eksin I. 一种新的优化方法: 大爆炸-大紧缩. *Adv Eng Softw* 2006;37:106–11.
- [24] Rashedi E, Nezamabadi-Pour H, Saryazdi S. GSA: 引力搜索算法. *Inf Sci* 2009;179:2232–48.
- [25] Kaveh A, Talatahari S. 新型启发式优化方法: 带电系统 search. *Acta Mech* 2010;213:267–89.
- [26] Formato RA. 中心力优化: 应用于应用电磁学的新型元追求. *Prog Electromag Res* 2007;77:425–91.
- [27] Alatas B. ACROA: 用于全局优化的人工化学反应优化算法. *Expert Syst Appl* 2011;38:13170–80.
- [28] Hatamlou A. Black hole: a new heuristic optimization approach for data clustering. *Inf Sci* 2012.
- [29] Kaveh A, Khayatizad M. 一种新的元启发式方法: 射线优化. *Comput Struct* 2012;112:283–94.
- [30] Du H, Wu X, Zhuang J. Small-world optimization algorithm for function optimization. In: Springer; 2006: Springer; 2006. p. 264–73.
- [31] Shah-Hosseini H. 基于星系搜索的主成分分析算法: 一种用于连续优化的新型元启发式. *Int J Comput Sci Eng* 2011;6:132–40.
- [32] Moghaddam FF, Moghaddam RF, Cheriet M. Curved space optimization: a random search based on general relativity theory. *ArXiv, preprint arXiv:1208.2214*; 2012.
- [33] Yang X-S. 一种新的元启发式蝙蝠算法. In: Springer; 2010: p. 65–74.
- [34] Abbass HA. MBO: 蜜蜂优化中的婚姻--一种合群多雌性蜂群方法. In: *Evolutionary computation*, 2001. p. 207–214.
- [35] Li X. 一种新的智能优化--人工鱼群算法. 中国浙江大学博士论文; 2003.
- [36] Roth M. Termite: a swarm intelligent routing algorithm for mobile wireless ad-hoc networks; 2005.
- [37] Pinto PC, Runkler TA, Sousa JM. 用于动态 MAX-SAT 问题的黄蜂群算法. In: *Adaptive and Natural Computing Algorithms*, ed... Springer; 2007: p. 350–57.
- [38] Mucherino A, Seref O. Monkey search: a novel metaheuristic search for global optimization. In: *AIP conference proceedings*; 2007. p. 162.
- [39] Lu X, Zhou Y. 新颖的全局收敛算法: 蜜蜂采集花粉算法. In: *高级智能计算理论与应用*. Springer; 2008: Springer; 2008. p. 518–25.
- [40] Yang X-S, Deb S. Cuckoo search via Lévy flights. In: *自然与生物启发计算*, 2009 年. NaBIC 2009. p. 210–14.
- [41] Y 世勤, Jianjun J, Y Guangxing Y. 海豚伙伴优化. In: *智能系统*, 2009 年. GCIS'09. WRI Global Congress on; 2009. p. 124–28.
- [42] Yang X-S. 萤火虫算法、随机测试函数与设计优化. *Int J Bio-Inspired Comput* 2010;2:78–84.
- [43] Askarzadeh A, Rezazadeh A. 新型质子交换膜燃料电池建模启发式优化算法: 鸟类交配优化器. *Int J Energy Res* 2012.
- [44] Gandomi AH, Alavi AH. Krill Herd: a new bio-inspired optimization algorithm. *Commun Nonlinear Sci Numer Simul* 2012.
- [45] Pan W-T. 一种新的果蝇优化算法: 以财务困境模型为例. *Knowl-Based Syst* 2012;26:69–74.
- [46] Mech LD. 狼群中的首领地位、优势和分工. *Can J Zool* 1999;77:1196–203.
- [47] Muro C, Escobedo R, Spector L, Coppinger R. 狼群 (Canis lupus) 狩猎策略源自计算模拟中的简单规则. *行为过程* 2011; 88:192–7.

- [48] Digalakis J, Margaritis K. 论遗传算法的基准函数.Int J Comput Math 2001;77:481-506.
- [49] Molga M, Smutnicki C. 优化需求的测试功能. 优化需求的测试功能》，2005 年。
- [50] Yang X-S. 优化中的测试问题，arXiv，preprint arXiv:1008.0549；2010.
- [51] Mirjalili S, Lewis A. 《二元粒子群优化的 S 型与 V 型传递函数》。Swarm Evolut Comput 2013;9:1-14.
- [52] Liang J, Suganthan P, Deb K. Newvel composition test functions for numerical global optimization.In: Swarm intelligence symposium, 2005.SIS 2005.Proceedings 2005 IEEE; 2005. p. 68-75.
- [53] Suganthan PN, Hansen N, Liang JJ, Deb K, Chen YP, Auger A 等人。《2005 年 CEC 实际参数优化特别会议的问题定义和评估标准》，技术报告，新加坡南洋理工大学，2005 年，http://www.ntu.edu.sg/home/EPNSugan。
- [54] van den Bergh F, Engelbrecht A. 粒子群优化粒子轨迹研究.Inf Sci 2006;176:937-71.
- [55] Arora JS. 最优设计导论》。学术出版社；2004 年。
- [56] Belegundu AD, Arora JS. 结构优化的数学编程方法研究。第一部分：理论.Int J Numer Meth Eng 1985;21:1583-99.
- [57] Coello Coello CA, Mezura Montes E. 通过使用基于优势的锦标赛选择来处理遗传算法中的约束条件。Adv Eng Inform 2002;16:193-203.
- [58] He Q, Wang L. 针对受限工程设计问题的有效协同进化粒子群优化.Eng Appl Artif Intell 2007;20:89-99.
- [59] Mezura-Montes E, Coello CAC. 关于进化策略在解决约束优化问题中的实用性的实证研究。Int J Gen Syst 2008;37:443-73.
- [60] Coello Coello CA. 在工程优化问题中使用自适应惩罚方法。Comput Ind 2000;41:113-27.
- [61] Mahdavi M, Fesanghary M, Damangir E. 用于解决优化问题的改进型和谐搜索算法. Appl Math Comput 2007;188:1567-79.
- [62] Huang F, Wang L, He Q. 用于约束优化的有效协同进化微分演化. Appl Math Comput 2007;186:340-56.
- [63] Yang XS. 自然启发的元启发式算法》。Luniver Press; 2011.
- [64] Carlos A, COELLO C. 利用进化多目标优化技术进行约束处理。Civil Eng Syst 2000;17:319-46.
- [65] Deb K. 通过遗传算法优化焊接梁的设计.AIAA J 1991;29:2013-5.
- [66] Deb K. 遗传算法的高效约束处理方法. Comput Methods Appl Mech Eng 2000;186:311-38.
- [67] Lee KS, Geem ZW. 用于连续工程优化的新元启发式算法：和谐搜索理论与实践. Comput Methods Appl Mech Eng 2005;194:3902-33.
- [68] Ragsdell K, Phillips D. 使用几何编程对一类焊接结构进行优化设计. ASME J Eng Indust 1976;98:1021-5.
- [69] Deb K, Gene AS. 用于机械部件设计的稳健优化设计技术。In: 提交给 Dasgupta D、Michalewicz Z 编辑。《工程应用中的进化算法》，柏林；1997 年。
- [70] Kaveh A, Talatahari S. 针对受限工程设计问题的改进型蚁群优化。Eng Comput Int J Comput-Aided Eng 2010;27:155-82.
- [71] Kannan B, Kramer SN. 基于增强拉格朗日乘法器的混合整数离散连续优化方法及其在机械设计中的应用。J Mech Des 1994;116:405.
- [72] Sandgren E. 机械设计中的非线性整数和离散编程；1988 年，第 95-105 页。
- [73] Caer C, Le Roux X, Marris-Morini D, Izard N, Vivien L, Gao D, et al. Photonics Technol Lett, IEEE 2011;23:1298-300.
- [74] Baba T. 光子晶体中的慢光。Nat Photonics 2008; 2:465-73.
- [75] Zhai Y, Tian H, Ji Y. 环形孔光子晶体波导的慢光特性改善与光缓冲能力. Light Technol J 2011; 29:3083-90.
- [76] Wang D, Zhang J, Yuan L, Lei J, Chen S, Han J, et al. Optics Commun 2011;284:5829-32.
- [77] Mirjalili SM, Mirjalili S. 在斜环形孔光子晶体波导中使用粒子群优化提高光特性和光缓冲器性能。In: p. 1-4 [2012].
- [78] Mirjalili SM, Abedi K, Mirjalili S. 利用粒子群优化提高环形孔光子晶体波导中的光缓冲性能。Optik - Int J Light Elect Optics 2013;124:5989-93.
- [79] Mirjalili SM, Mirjalili S, Lewis A. 设计光子晶体波导的新型多目标优化框架。Photonics Technol Lett IEEE 2014;26:146-9.
- [80] Mirjalili SM, Mirjalili S, Lewis A, Abedi K. A tri-objective particle swarm optimizer for designing line defect photonic crystal waveguides. 光子学与纳米结构--基础与应用》。
- [81] Wu J, Li Y, Peng C, Wang Z. 槽型光子晶体波导中的宽带低色散慢速光. Optics Commun 2010; 283:2815-9.
- [82] Mirjalili S, Mirjalili SM, Yang X. 二进制蝙蝠算法。神经计算应用》，出版中，DOI: 10.1007/s00521-013-1525-5。