
文献阅读：
多目标灰狼优化算法：
一种多准则优化的新算
法

2022.11.13

摘 要

由于灰狼优化算法（GWO）的新颖性，文献中还没有设计该算法的多目标版本的研究。针对多目标优化问题，本文首次提出了一种多目标灰狼优化（MOGWO）算法。一个规模不变的外部存档被集成到 GWO 中，用于保存和检索帕累托最优解。然后，该存档被用来定义社会等级，并模拟灰狼在多目标搜索空间中的狩猎行为。该方法在 10 个多目标基准问题上进行了测试，并与两个著名的元启发式算法——基于分解的多目标进化算法（MOEA/D）和多目标粒子群优化（MOPSO）算法进行了比较。定性和定量的结果表明，所提出的算法能够给出非常有竞争力的结果，并优于其他算法。请注意，MOGWO 的源代码在 <http://www.alimirjalili.com/GWO.html> 上公开提供。

关键字：多目标优化；进化算法；多准则优化；启发式算法；元启发式算法；工程优化；灰狼优化算法

目 录

摘 要	2
目 录	3
第一章 引言	4
第二章 文献综述	5
2.1 多目标优化	5
2.2 多目标元启发式算法	6
第三章 多目标灰狼优化算法 (MOGWO)	8
第四章 结果与讨论	13
4.1 实验设置	13

第一章 引言

在求解实际工程问题时存在许多挑战，这需要特定的工具来处理它们。现实问题的一个最重要的特点是多目标性，这使得它们具有挑战性。如果有一个以上的目标需要优化，则该问题就被称为多目标优化问题。不用说，应采用多目标优化算法来求解这些问题。有两种处理多目标的方法：先验(priori)与后验(posteriori) (Branke, Kaußler, & Schmeck, 2001; Marler & Arora, 2004)。

前一类优化方法将多目标问题的目标组合成一组具有权重（由决策者提供）的单个目标，权重定义了各个目标的重要程度，并采用单目标优化算法来进行求解。组合搜索空间的单目标特性允许找到单个解作为最优解。相反，后验方法保持了多目标问题的多目标表述，与先验方法相比，允许在一系列设计参数和运行条件下探索问题的行为 (Deb, 2012)。在这种情况下，决策者最终会根据自己的需求选择所获得的解中的一个。还有另一种处理多目标的方法，叫做渐进法，在优化过程中考虑决策者对目标的偏好 (Branke & Deb, 2005)。

与单目标优化不同，当考虑多个目标作为优化过程的目标时，不会出现单一解的现象。在这种情况下，表示目标之间各种权衡的一组解构成了多目标问题的最优解 (Coello, Lamont, & Van Veldhuisen, 2007)。1984 年以前，数学多目标优化技术在不同研究领域中很受欢迎，如应用数学、运筹学和计算机科学。然而，由于大多数传统方法（包括确定性方法）都存在局部最优的停滞现象，因此此类技术现在还没有实用的方法。

1984 年，David Schaffer 提出了一个革命性的想法 (Coello Coello, 2006)。他利用随机优化技术（包括进化算法和启发式算法）提出了多目标优化的概念。令人惊讶的是，从那时起，大量的研究都致力于开发和评估多目标进化/启发式算法。随机优化技术的优势，如无梯度机制和避免局部优化，使它们很容易应用于实际问题。如今，多目标优化技术的应用可以在不同的研究领域中找到：机械工程 (Kipouros 等, 2008)、土木工程 (Luh & Chueh, 2004)、化学 (Gaspar-Cunha & Covas, 2004; Rangaiah, 2008) 以及其他领域 (Coello & Lamont, 2004)。

在多目标随机优化的早期，人们将单目标优化算法转换为多目标优化算法。迄今提出的一些最著名的随机优化技术如下：

- Strength-Pareto Evolutionary Algorithm (SPEA) (Zitzler, 1999; Zitzler & Thiele, 1999);
- 非支配性排序遗传算法 (Srinivas & Deb, 1994);
- 非支配性排序遗传算法 II (NSGA-II) (Deb, Pratap, Agarwal, & Meyarivan, 2002);
- 多目标粒子群优化算法 (MOPSO) (Coello, Pulido, & Lechuga, 2004);
- 基于分解的多目标进化算法 (MOEA/D) (Zhang & Li, 2007);
- Pareto Archived Evolution Strategy (PAES) (Knowles & Corne, 2000);
- Pareto-frontier Differential Evolution (PDE) (Abbass, Sarker, & Newton, 2001)。

文献表明，这些算法能够有效地接近多目标问题的真实帕累托最优解。然而，“没有免费的午餐” (NFL) 定理 (Wolpert & Macready, 1997) 从逻辑上证明了没有能求解所有优化问题的优化技术。根据该定理，一个优化方法在一类问题上的优越表现不能保证在另一类问题上也有类似的表现。该定理是文献中许多工作的基础，使该领域的研究人员能够将目前的技术用于新的问题类型或提出新的优化算法。这也是这项工作的基础和动机，我们在最近提出的灰狼优化 (GWO) 算法

的基础上，提出了一种新的多目标优化算法，称为多目标灰狼优化（MOGWO）算法。这项研究的贡献如下：

- 在 GWO 算法中集成了一个存档（archive），用于储存非支配性解。
- 一个网格（grid）机制已被整合到 GWO 中，以改善存档中的非支配性解。
- 提出了一种基于 alpha、beta 和 delta 狼的领导者选择机制，以更新和替换存档中的解。
- 利用上述三种算子提出了 GWO 的多目标版本。

本文的其余部分组织如下：第二章介绍了多目标搜索空间中优化的定义和预备知识。第三章简要回顾了 GWO 的概念，然后提出了 MOGWO 算法。第四章中介绍了定性和定量结果以及相关的讨论。最后，第五部分总结了本文的工作，并对未来的工作提出了一些建议。

第二章 文献综述

本章给出了多目标优化的概念和现有的元启发式算法。

2.1 多目标优化

正如在引言中简要提到的，多目标优化是指对一个具有多个目标函数的问题进行优化。为不失一般性，它可以表述为如下的最大化问题：

$$\text{Maximize : } (F(\vec{x}) = f_1(\vec{x}), f_2(\vec{x}), \dots, f_o(\vec{x})) \quad (2.1)$$

$$\text{Subject to : } (g_i(\vec{x}) \geq 0, i = 1, 2, \dots, m) \quad (2.2)$$

$$h_i(\vec{x}) = 0, i = 1, 2, \dots, p \quad (2.3)$$

$$L_i \leq x_i \leq U_i, i = 1, 2, \dots, n \quad (2.4)$$

其中， n 为变量个数， o 为目标函数数量， m 为不等式约束个数， p 为等式约束个数， g_i 为第 i 个不等式约束， h_i 为第 i 个等式约束， $[L_i, U_i]$ 为第 i 个变量的边界。

在单目标优化中，由于目标函数只有一个，所以可以很容易地对解进行比较。对于最大化问题，当且仅当 $X > Y$ 时，解 X 优于 Y 。然而，由于多准则比较指标的存在，多目标空间中的解无法通过关系运算符进行比较。在这种情况下，一个解优于（支配）另一个解，当且仅当它在所有目标上表现出更好或相等的目标值，并在至少一个目标函数中给出更好的值。多目标问题中两个解的比较概念首先由 Francis Ysidro（Edgeworth, 1881）提出，然后由 Vilfredo Pareto（Pareto, 1964）扩展。不失一般性，最大化问题的帕累托支配的数学定义如下（Coello, 2009）：

定义 1. 帕累托支配：

假设有两个向量，如 $\vec{x} = (x_1, x_2, \dots, x_k)$ 和 $\vec{y} = (y_1, y_2, \dots, y_k)$ 。

向量 x 支配向量 y （表示为 $x \succ y$ ），当且仅当：

$$\forall i \in \{1, 2, \dots, k\}, [f(x_i) \geq f(y_i)] \wedge [\exists i \in \{1, 2, \dots, k\} : f(x_i)] \quad (2.5)$$

帕累托最优的定义如下 (Ngatchou, Zarei, & El-Sharkawi, 2005):

定义 1. 帕累托最优:

一个解 $\bar{x} \in X$ 被称为帕累托最优, 当且仅当:

$$\nexists \bar{y} \in X \mid F(\bar{y}) \succ F(\bar{x}) \quad (2.6)$$

包括问题的所有非支配性解的集合被称为帕累托最优集, 其定义如下:

定义 3. 帕累托最优集:

所有帕累托最优解的集合被称为帕累托最优集, 如下所示:

$$P_s := \{x, y \in X \mid \exists F(y) \succ F(x)\} \quad (2.7)$$

在帕累托最优集中包含帕累托最优解的相应目标值的集合称为帕累托最优前沿 (front)。帕累托最优前沿的定义如下:

定义 4. 帕累托最优前沿: 包含帕累托最优集的目标函数值的集合。

$$P_f := \{F(x) \mid x \in P_s\} \quad (2.8)$$

2.2 多目标元启发式算法

多目标优化算法 (后验方法) 的最终目标是找到具有最高多样性的真实帕累托最优解的非常精确的近似 (Zhou 等人, 2011)。这使得决策者可以有多样化的设计方案。在过去, 多目标问题的求解是通过先验地将目标组合为一个单一目标来进行的。然而, 这种方法有两个主要缺点 (Das & Dennis, 1998; Kim & De Weck, 2005; Messac & Mattson, 2002)。权重的均匀分布不一定保证找到具有均匀分布的帕累托最优解, 而且该方法不能找到帕累托最优前沿的非凸区域, 因为不允许负权重, 所有的权重之和应为常数。换句话说, 在传统的聚合 (aggregation) 方法中, 通常使用目标的凸和 (convex sum)。

在文献中, 有一些作品试图改进这种方法。例如, Parsopoulos 和 Vrahatis 使用两种动态加权聚合函数, 随时间逐渐或突然改变权重 (Parsopoulos & Vrahatis, 2002)。然而, 这些方法没有完全解决聚合方法的主要问题。此外, 这些方法需要运行多次才能接近整个帕累托最优解, 因为每次迭代只得到一个最优解。根据 Deb (Deb, 2012), 利用元启发式方法的多目标优化过程涉及克服许多困难, 如不可行区域、局部前沿、解的多样性和最优解的分隔 (isolation)。先验方法应在每次运行中处理所有这些困难。然而, 保持问题的多目标表述会带来一些好处。首先, 关于搜索空间的信息在搜索智能体之间进行交换, 这导致向真实帕累托最优前沿的快速移动。

其次, 多目标方法有助于在一次运行中接近整个真实帕累托最优前沿。最后, 保持问题的多目标表述允许在一系列设计参数和运行条件下探索问题的行为。然而, 先验方法的唯一缺点是需要使用更复杂的元启发式方法和解决矛盾的目标。几乎大多数最著名的启发式算法已经被扩展到解决多目标问题。下文我们将简要介绍最流行和最新的算法。

文献表明, 最流行的多目标元启发式算法是非支配性排序遗传算法 II (NSGA-II) (Deb 等人, 2002), 它是广受好评的 GA 算法的多目标版本 (Goldberg,

1989; Goldberg & Holland, 1988)。这个算法是为了缓解 NSGA 的三个问题而提出的 (Srinivas & Deb, 1994)。这些问题是非支配性排序的高计算成本, 没有考虑精英机制, 以及缺乏共享参数 (不同于小生境 (niching))。为了缓解上述问题, NSGA-II 采用了快速非支配性排序技术、精英保留技术和新的无参数小生境 (niching) 算子。

NSGA-II 算法从随机种群开始。个体根据非支配性排序方法进行分组。每个个体的适应度是根据其非支配性水平来定义的。第二个种群是由选择、交叉和变异算子生成的。两个种群组合成一个新的大种群。然后, 这个新种群通过非支配性排序方法再次进行排序。非支配性等级越高, 选择作为最终种群的新个体的优先级就越高。重复选择非支配性个体, 直到具有与初始种群相同规模的种群。最后, 执行这些步骤, 直到满足终止标准。

第二个最流行的多目标元启发式算法是多目标粒子群优化 (MOPSO) 算法。MOPSO 算法是由 Coello 等人, 2004 年; Coello Coello 和 Lechuga, 2002 年提出的。遵循与 PSO 相同的概念, 它采用大量的粒子, 这些粒子在搜索空间中飞行来寻找最好的解。同时, 它们都在自己的路径上追踪最好的位置 (最好的解) (Shi & Eberhart, 1998)。与 PSO 相比, 当然没有单个“最好的”解可以追踪。换句话说, 粒子在更新位置时, 必须考虑它们自身的非支配性解 (*pbest*), 以及粒子群到目前为止获得的非支配性解之一 (*gbest*)。通常使用一个外部存档来存储和检索所获得的帕累托最优解。此外, 在 MOPSO 中偶尔会加入一种叫做湍流 (turbulence) 的变异操作, 以增加随机性并促进试验解的多样性。对基于 PSO 的多目标优化算法的综述可以在 Reyes-Sierra 和 Coello, 2006 中找到。

提出的外部存档与 Pareto Archived Evolution Strategy (PAES) (Knowles & Corne, 2000) 中的自适应网格 (grid) 类似, 因为它被设计成保存到目前为止获得的非支配性解。它有两个主要组成部分: 存档控制器 (archive controller) 和网格。前一个负责决定一个解是否应该被添加到存档中。如果一个新解被存档中的某个个体所支配, 则它应被立即忽略掉。如果新解没有被存档中的个体所支配, 则应该将其添加到存档中。如果存档中的某个个体被新的解所支配, 则它就被新的解替换掉。最后, 如果存档已满, 则触发自适应网格机制。

网格部分负责保持存档解尽可能的多样化。在这种方法中, 目标空间被划分为几个区域。如果一个新获得的解位于网格之外, 所有的网格位置都要重新计算以覆盖这个解。如果一个新解位于网格内, 它将被引导到网格中粒子数量最少的区域。与小生境相比, 这种网格的主要优点是计算成本低 (在最坏的情况下, 当网格必须在每次迭代中更新时, 它与小生境 $O(N^2)$ 相同)。

MOPSO 具有非常快的收敛速度, 这可能使其在多目标优化中容易过早终止, 出现假帕累托最优前沿 (Nebro, Durillo, & Coello, 2013)。变异策略在这种情况下是有帮助的。它不仅随机地影响粒子群中的粒子, 还随机影响设计变量。变异概率在迭代过程中逐渐降低。

MOPSO 算法首先在问题空间中随机放置粒子。在迭代过程中, 计算粒子的速度。定义了速度之后, 粒子位置可以被更新。所有非支配性解被添加到存档中。最后, 搜索过程在满足终止条件后终止。

在过去的三年内, 又提出了许多多目标优化算法。多目标猫群优化 (MOCSO) 算法 (Pradhan & Panda, 2012), 多目标蚁群优化 (Shi & Kong, 2015) 算法, 多目标教与学优化算法 (Lin et al, 2015), 多目标人工蜂群算法 (Hancer, Xue, Zhang,

Karaboga, & Akay, 2015), 多目标差分进化算法 (Osorio Velazquez, Coello Coello, & Arias-Montano, 2014), 以及多目标引力搜索算法 (MOGSA) (Hemmatian, Fereidoon, & Assareh, 2014)。这些研究显示了元启发式算法处理多目标的能力。尽管以上讨论的所有算法都能够近似于给定问题的真实帕累托最优前沿, 但是根据 NFL 定理, 它们不能解决所有优化问题。因此, 很可能一种新算法解决了文献中现有技术无法解决的问题。在下面的章节中, 我们提出了一种新的多目标 GWO, 作为目前文献中解决多目标优化问题的算法的替代方案。

第三章 多目标灰狼优化算法 (MOGWO)

GWO 算法是由 Mirjalili, Mirjalili, and Lewis, 2014 年提出的。灰狼的社会领导和狩猎技术是该算法的主要灵感来源。为了在设计 GWO 时对狼的社会等级进行数学建模, 最合适的解被认为是 α (α) 狼。因此, 第二和第三最好的解分别被命名为 β (β) 和 δ (δ) 狼。其余的候选解被认为是 ω (ω) 狼。在 GWO 算法中, 狩猎 (优化) 由 α 、 β 和 δ 指导。 ω 狼跟随这三只狼寻找全局最优。

除了社会领导之外, 为了模拟灰狼在狩猎过程中的包围行为, 还提出了以下公式 (图 3) (Mirjalili 等, 2014):

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

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

其中 t 为当前迭代次数, \vec{A} 和 \vec{C} 为系数向量, \vec{X}_p 为猎物的位置向量, \vec{X} 表示灰狼的位置向量。

向量 \vec{A} 和 \vec{C} 的计算方法如下:

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \quad (3.3)$$

$$\vec{C} = 2 \cdot \vec{r}_2 \quad (3.4)$$

其中 \vec{a} 的元素在迭代过程中从 2 线性减小到 0, r_1 、 r_2 是 [0,1] 内的随机向量。如图 1 所示。

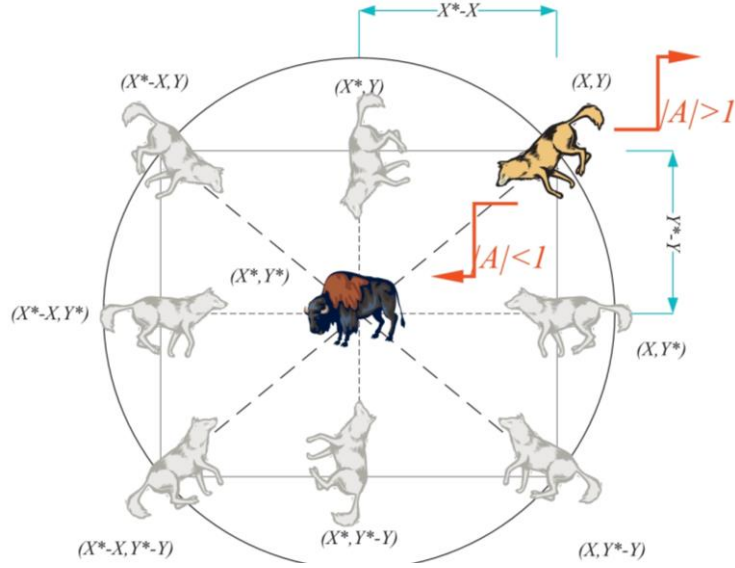


图 1 搜索智能体的位置更新机制及 A 对其影响

GWO 算法利用模拟的社会领导和包围机制来寻找优化问题的最优解。该算法保存了到目前为止获得的前三个最好的解，并迫使其他搜索智能体（包括 ω ）更新它们的相对位置。在优化过程中，每个搜索智能体不断运行以下公式，以模拟狩猎并找到搜索空间中有希望的区域：

$$\vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}| \quad (3.5)$$

$$\vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}| \quad (3.6)$$

$$\vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}| \quad (3.7)$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot (\vec{D}_\alpha) \quad (3.8)$$

$$\vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot (\vec{D}_\beta) \quad (3.9)$$

$$\vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot (\vec{D}_\delta) \quad (3.10)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (3.11)$$

探索由随机值大于 1 或小于 -1 的 \vec{A} 来保证，它迫使搜索智能体远离猎物。

GWO 中另一个有利于探索的元素是 \vec{C} 。向量 \vec{C} 生成 [0, 2] 内的随机数，其中提供了猎物的随机权重，以便随机强调 ($C > 1$) 或削弱 ($C < 1$) 猎物在定义公式 (3.1) 中距离的影响。这有助于 GWO 在整个优化过程中表现得更加随机，有利于探索和避免陷入局部最优。值得一提的是，与 A 不同， C 不是线性减小的。 C 参数被要求始终提供随机值，以便不仅在迭代初始期间，而且在最后的迭代期间强调探索。这个元素在陷入局部最优的情况下非常有用，尤其是在最后的迭代中。

当 $|A| < 1$ 时，GWO 算法的开发（exploitation）就开始了。当 \bar{A} 的随机值在 $[-1, 1]$ 时，搜索智能体的下一个位置可以在其当前位置和猎物位置之间的任意位置，这有助于搜索智能体向 α 、 β 和 δ 解给出的猎物的估计位置收敛。

GWO 算法开始优化时，生成一组随机解作为初始种群。在优化过程中，保存下到目前为止获得的三个最好的解，称为 α 、 β 和 δ 解。对于每只 ω 狼（除 α 、 β 和 δ 之外的搜索智能体），触发位置更新公式（3.5）至（3.11）。同时，参数 a 和 A 在迭代过程中线性减小。因此，当 $|\bar{A}| > 1$ 时，搜索智能体倾向于偏离猎物，

而当 $|\bar{A}| < 1$ 时，则倾向于向猎物收敛。最后，当满足一个终止条件时， α 解的位置和适应度值作为整个优化过程中获得的最优解返回。

为了将 GWO 用于多目标优化，我们集成了两个新组件。采用的组件与 MOPSO（Coello 等人，2004）的组件非常相似。其一是存档，负责存储到目前为止获得的非支配性帕累托最优解。其二是一个领导者选择策略，协助选择 α 、 β 和 δ 的解，作为从存档中获得的狩猎过程的领导者。

存档是一个简单的存储单元，可以保存或检索到目前为止获得的非支配性帕累托最优解。存档的关键模块是一个存档控制器，它在解想进入存档或存档已满时控制存档。请注意，存档有一个最大个体数量。在迭代的过程中，到目前为止获得的非支配性解与存档中的个体进行比较。会有以下三种不同的情况：

- 新个体被至少一个存档中的个体所支配。在这种情况下，不允许该解进入存档。
- 新解支配了存档中一个或多个解。在这种情况下，忽略掉存档中被支配的解，而新解将进入存档。
- 如果新解和存档中的个体都没有互相支配，则应当将新解添加到存档中。
- 如果存档已满，应首先运行网格机制，重新安排目标空间的划分，并找到最拥挤的区域，忽略掉其中的一个解。然后，将新解插入到最不拥挤的区域，以提高最终近似帕累托最优前沿的多样性。

删除一个解的概率与超立方体（区域）中解的数量成正比。如果存档已满，为了删除解，首先选择最拥挤的区域，并从其中一个区域中随机忽略掉一个解，以便为新解提供空间。有一种特殊情况，在超立方体外插入一个解。在这种情况下，为了覆盖新解，所有的区域都被扩展。因此，其他解的所属区域也可以被改变。

第二个组成部分是领导者选择机制。在 GWO 中，到目前为止获得的三个最好的解被视为 α 狼、 β 狼和 δ 狼。这些领导者引导其他搜索智能体走向搜索空间中有希望的区域，希望能找到近似全局最优解。然而，在多目标搜索空间中，由于前面小节中讨论的帕累托最优的概念，不能轻松地比较解。领导者选择机制就是为了处理这个问题而设计的。如上所述，有一个到目前为止获得的最好的非支配性解的存档。领导者选择组件选择搜索空间中最不拥挤的区域，并给出其非支配性解中的一个作为 α 、 β 或 δ 狼。通过轮盘赌方法进行选择，每个超立方体的概率如下：

$$p_i = \frac{c}{N_i} \quad (3.12)$$

其中 c 为一个大于 1 的常数, N_i 是第 i 个区域中获得的帕累托最优解的数量。

从公式 (3.11) 中可以看出, 不那么拥挤的超立方体有更高的概率给出新的领导者。当超立方体中获得的解的数量减少时, 选择超立方体来选择领导者的概率会增加。应当注意的是, 可能有一些特殊情况, 因为我们必须选择三个领导者。如果在最不拥挤的区域有三个解, 其中三个被随机分配为 **alpha**、**beta** 和 **delta** 解。如果在最不拥挤的超立方体中少于三个解, 也会找到第二个最不拥挤的超立方体来选择其他的领导者。如果第二个最不拥挤的超立方体有一个解, 这种情况也是一样的, 因此应该从第三个最不拥挤的超立方体中选择 **delta** 领导者。通过这种方法, 我们防止 MOGWO 为 **alpha**、**beta** 或 **delta** 挑选类似的领导者。因此, 搜索总是朝着搜索空间中未开发/未暴露的区域进行, 因为领导者选择机制倾向于最不拥挤的超立方体, 如果在最不拥挤的区域没有足够数量的领导者(少于 3 个), 则提供来自不同区域的领导者。

MOGWO 的计算复杂度为 $O(MN^2)$, 其中 N 为种群规模, M 为目标个数。

该复杂度与该领域的其他著名算法相当。NSGA-II (Deb 等人, 2002), MOPSO, SPEA2 (Zitzler 等人, 2001), 和 PAES (Knowles & Corne, 1999)。然而, 其计算复杂度优于一些算法, 如 NSGA (Srinivas & Deb, 1994) 和 SPEA (Zitzler & Thiele, 1998), 后者为 $O(MN^3)$ 。

MOGWO 算法的伪代码在图 2 中给出。

初始化灰狼种群 $X_i (i = 1, 2, \dots, n)$

初始化 a 、 A 和 C

计算每个搜索智能体的目标函数值

找到非支配性解，并用它们对存档进行初始化

$X_\alpha = \text{SelectLeader}(\text{archive})$

暂时将 α 排除在存档之外，以避免选择相同的领导者

$X_\beta = \text{SelectLeader}(\text{archive})$

暂时将 β 排除在存档之外，以避免选择相同的领导者

$X_\delta = \text{SelectLeader}(\text{archive})$

将 α 和 β 添加回存档

$t = 1$;

while ($t < \text{Max number of iterations}$)

for 每个搜索智能体

通过公式 (3.5) - (3.11) 更新当前搜索智能体的位置

end for

更新 a 、 A 和 C

计算每个搜索智能体的目标函数值

找到非支配性解

根据所获得的非支配性解更新存档

If 存档已满

运行网格机制，忽略掉当前的一个存档个体

将新解添加到存档中

end if

If 任意新加入存档的解位于超立方体之外

更新网格以覆盖新的解

end if

$X_\alpha = \text{SelectLeader}(\text{archive})$

暂时将 α 排除在存档之外，以避免选择相同的领导者

$X_\beta = \text{SelectLeader}(\text{archive})$

暂时将 β 排除在存档之外，以避免选择相同的领导者

$X_\delta = \text{SelectLeader}(\text{archive})$

将 α 和 β 添加回存档

$t = t + 1$

end while

return archive

图 2 MOGWO 算法的伪代码

为了解所提出的 MOGWO 算法如何在理论上有效地解决多目标问题，可以注意如下一些评价：

- 所采用的外部存档有效地保存了到目前为止获得的最好的非支配性解。
- 由于 MOGWO 继承了 GWO 的包围机制，在解周围有一个圆形的邻域，它可以扩展到更高维度的超球体（在参数空间）。
- 随机参数 A 和 C 赋予候选解不同随机半径的超球体。
- 由于 MOGWO 继承了 GWO 的狩猎机制，搜索智能体被允许定位猎物的可能位置。
- 探索 and 开发是由 a 和 A 的自适应值保证的。
- 参数 a 和 A 的自适应值使 MOGWO 能够在探索 and 开发之间平稳过渡。因此，MOGWO 算法的收敛性得到了保证。
- 随着 A 的减小，一半的迭代用于探索（ $|A| \geq 1$ ），另一半用于开发（ $|A| < 1$ ）。
- MOGWO 只有两个关键参数需要调整（ a 和 C ）。
- 网格机制和选择领导者组件在优化过程中保证了存档的多样性。
- 领导者选择组件中采用的轮盘赌法给出了从最拥挤的超立方体中选择领导者的低概率。这就避免了 MOGWO 陷入局部前沿。
- 在优化过程中，参数 C 的非自适应性随机值也同时促进了 MOGWO 算法的探索 and 避免入局部前沿。

这里需要指出的是，MOGWO 算法的收敛性是有保证的，因为它利用了相同的数学模型来搜索最优解。事实证明，GWO 要求搜索智能体在优化的初期突然改变位置，在优化的后期逐渐改变位置（Mirjalili 等人，2014）。根据 Van den Bergh 和 Engelbrecht, 2006，这种行为保证了算法在搜索空间的收敛性。MOGWO 算法继承了 GWO 的所有特征，这意味着搜索智能体以相同的方式探索 and 开发搜索空间。主要的区别是，MOGWO 围绕一组存档个体进行搜索（即使存档没有变化，也可能不同），而 GWO 只保存和改进三个最好的解。

该算法与最近一个同类算法，Pradhan 和 Panda 于 2012 年提出的多目标猫群优化（MOCSSO）算法之间的区别是，MOCSSO 同时采用了非支配性排序和存档，而 MOGWO 只采用外部存档来存储非支配性解。

第四章 结果与讨论

本章概述了实验设置，给出了研究结果，并进行了讨论。

4.1 实验设置

多目标优化算法（后验方法）的最终目标是找到具有最高多样性的真实帕累托最优解的非常精确的近似（Zhou 等人，2011）。这使得决策者可以有多样化的设计方案。在过去，

将 MOGWO 算法与文献中的两个著名算法进行比较：MOPSO 和 MOEA/D。对于 MOPSO，选择以下初始参数：

- $\varphi_1 = \varphi_2 = 2.05$
- $\varphi = \varphi_1 + \varphi_2$

- $w = \chi = \frac{2}{\varphi - 2 + \sqrt{\varphi^2 - 4\varphi}}$ ：惯性权重
 - $c_1 = \chi * \varphi_1$ ：个体因子
 - $c_2 = \chi * \varphi_2$ ：社会因子
 - $\alpha = 0.1$ ：网格膨胀因子
 - $\beta = 4$ ：领导者选择压力因子
 - $nGrid = 10$ ：每一维上的网格数量
- 对于 MOEA/D，选择以下初始参数：
- $N = 100$ ：子问题
 - $T = 0.1N = 10$ ：邻居数量
 - $n_r = 0.01N = 1$ ：更新中的一个子代的最大副本（copies）
 - $\delta = 0.9$ ：从邻居中选择亲本的概率
 - $CR = F = 0.5$ ：变异概率
 - $\eta = 30$ ：分布指数

请注意，在所有的实验中，我们使用了 100 个搜索智能体和最多 3000 次迭代。作为测试平台，我们选择了 CEC 2009 中提出的 10 个标准多目标测试问题（Zhang 等人，2008）。表 1 和表 2 中给出了基准问题。这些测试问题被认为是文献中最具挑战性的测试问题，它们提供了不同的多目标搜索空间和帕累托最优前沿：凸、非凸、不连续和多模态的。

对于性能指标（Fonseca, Knowles, Thiele, & Zitzler, 2005），我们使用反世代距离（IGD）来衡量收敛性。采用间距（SP）（Carlos A Coello Coello 等人，2004；Schott, 1995）和最大分散度（MS）（Zitzler, 1999）来量化和衡量覆盖率。IGD 的数学表述与世代距离（GD）（Van Veldhuizen & Lamont, 1998）相似。这一改进后的衡量方法是由 Sierra 和 Coello 提出的，表述如下：

$$IGD = \frac{\sqrt{\sum_{i=1}^n d_i^2}}{n} \quad (4.1)$$

其中 n 为真实帕累托最优解的数量， d_i 表示第 i 个真实帕累托最优解与参考集中最近的获得的帕累托最优解间的欧氏距离。在这里，获得的解和参考集之间的欧氏距离是不同的。在 IGD 中，每一个真实解的欧氏距离是相对于它在目标空间中最接近的获得的帕累托最优解计算的。

SP 和 MS 方法的数学表述如下：

$$SP = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\bar{d} - d_i)^2} \quad (4.2)$$

其中， \bar{d} 为 d_i 的平均值， n 为获得的帕累托最优解的数量，

$$d_i = \min_j \left(\left| f_1^i(\bar{x}) - f_1^j(\bar{x}) \right| + \left| f_2^i(\bar{x}) - f_2^j(\bar{x}) \right| \right), i, j = 1, 2, 3, \dots, n$$

$$MS = \sqrt{\sum_{i=1}^o \max(d(a_i, b_i))} \quad (4.3)$$

其中 d 为计算欧氏距离的函数， a_i 为第 i 个目标中的最大值， b_i 为第 i 个目标中的最小值， o 为目标数量。

除了利用上述性能指标，使我们能够定量比较 MOGWO 与 MOPSO、MOEA/D，我们还说明了每个算法在参数空间和搜索空间获得的最好的帕累托最优解集。这也能使我们定性比较算法的性能。所有的算法在测试问题上运行了 10 次，表 3-5 中给出了这 10 次运行的统计结果。注意，我们对每个算法使用了 300,000 次函数评估，每个测试函数的参数数量为 30。图 3-5 中也给出了定性结果。

表 1 双目标测试问题

Name	Mathematical formulation
UF1	$f_1 = x_1 + \frac{2}{ J_1 } \sum_{j \in J_1} [x_j - \sin(6\pi x_1 + \frac{j\pi}{n})]^2$, $f_2 = 1 - \sqrt{x_1} + \frac{2}{ J_2 } \sum_{j \in J_2} [x_j - \sin(6\pi x_1 + \frac{j\pi}{n})]^2$
UF2	$J_1 = \{j j \text{ is odd and } 2 \leq j \leq n\}$, $J_2 = \{j j \text{ is even and } 2 \leq j \leq n\}$ $f_1 = x_1 + \frac{2}{ J_1 } \sum_{j \in J_1} y_j^2$, $f_2 = 1 - \sqrt{x_1} + \frac{2}{ J_2 } \sum_{j \in J_2} y_j^2$
UF3	$J_1 = \{j j \text{ is odd and } 2 \leq j \leq n\}$, $J_2 = \{j j \text{ is even and } 2 \leq j \leq n\}$ $y_j = \begin{cases} x_j - [0.3x_1^2 \cos(24\pi x_1 + \frac{4j\pi}{n}) + 0.6x_1] \cos(6\pi x_1 + \frac{j\pi}{n}) & \text{if } j \in J_1 \\ x_j - [0.3x_1^2 \cos(24\pi x_1 + \frac{4j\pi}{n}) + 0.6x_1] \sin(6\pi x_1 + \frac{j\pi}{n}) & \text{if } j \in J_2 \end{cases}$ $f_1 = x_1 + \frac{2}{ J_1 } (4 \sum_{j \in J_1} y_j^2 - 2 \prod_{j \in J_1} \cos(\frac{20y_j\pi}{\sqrt{j}}) + 2)$, $f_2 = \sqrt{x_1} + \frac{2}{ J_2 } (4 \sum_{j \in J_2} y_j^2 - 2 \prod_{j \in J_2} \cos(\frac{20y_j\pi}{\sqrt{j}}) + 2)$
UF4	J_1 and J_2 are the same as those of UF1, $y_j = x_j - x_1^{0.5(1.0 + \frac{3(j-2)}{n-2})}$, $j = 2, 3, \dots, n$ $f_1 = x_1 + \frac{2}{ J_1 } \sum_{j \in J_1} h(y_j)$, $f_2 = 1 - x_2 + \frac{2}{ J_2 } \sum_{j \in J_2} h(y_j)$
UF5	J_1 and J_2 are the same as those of UF1, $y_j = x_j - \sin(6\pi x_1 + \frac{j\pi}{n})$, $j = 2, 3, \dots, n$, $h(t) = \frac{ t }{1+e^{2 t }}$ $f_1 = x_1 + (\frac{1}{2N} + \epsilon) \sin(2N\pi x_1) + \frac{2}{ J_1 } \sum_{j \in J_1} h(y_j)$, $f_2 = 1 - x_1 + (\frac{1}{2N} + \epsilon) \sin(2N\pi x_1) + \frac{2}{ J_2 } \sum_{j \in J_2} h(y_j)$
UF6	J_1 and J_2 are identical to those of UF1, $\epsilon > 0$, $y_j = x_j - \sin(6\pi x_1 + \frac{j\pi}{n})$, $j = 2, 3, \dots, n$ $h(t) = 2t^2 - \cos(4\pi t) + 1$ $f_1 = x_1 + \max\{0, 2(\frac{1}{2N} + \epsilon) \sin(2N\pi x_1)\} + \frac{2}{ J_1 } (4 \sum_{j \in J_1} y_j^2 - 2 \prod_{j \in J_1} \cos(\frac{20y_j\pi}{\sqrt{j}}) + 1)$ $f_2 = 1 - x_1 + \max\{0, 2(\frac{1}{2N} + \epsilon) \sin(2N\pi x_1)\} + \frac{2}{ J_2 } (4 \sum_{j \in J_2} y_j^2 - 2 \prod_{j \in J_2} \cos(\frac{20y_j\pi}{\sqrt{j}}) + 1)$
UF7	J_1 and J_2 are identical to those of UF1, $\epsilon > 0$, $y_j = x_j - \sin(6\pi x_1 + \frac{j\pi}{n})$, $j = 2, 3, \dots, n$ $f_1 = \sqrt[3]{x_1} + \frac{2}{ J_1 } \sum_{j \in J_1} y_j^2$, $f_2 = 1 - \sqrt[3]{x_1} + \frac{2}{ J_2 } \sum_{j \in J_2} y_j^2$ J_1 and J_2 are identical to those of UF1, $\epsilon > 0$, $y_j = x_j - \sin(6\pi x_1 + \frac{j\pi}{n})$, $j = 2, 3, \dots, n$

表 2 三目标测试问题

Name	Mathematical formulation
UF8	$f_1 = \cos(0.5x_1\pi) \cos(0.5x_2\pi) + \frac{2}{ J_1 } \sum_{j \in J_1} (x_j - 2x_2 \sin(2\pi x_1 + \frac{j\pi}{n}))^2$ $f_2 = \cos(0.5x_1\pi) \sin(0.5x_2\pi) + \frac{2}{ J_2 } \sum_{j \in J_2} (x_j - 2x_2 \sin(2\pi x_1 + \frac{j\pi}{n}))^2$ $f_3 = \sin(0.5x_1\pi) + \frac{2}{ J_3 } \sum_{j \in J_3} (x_j - 2x_2 \sin(2\pi x_1 + \frac{j\pi}{n}))^2$ $J_1 = \{j 3 \leq j \leq n, \text{ and } j-1 \text{ is a multiplication of } 3\}, J_2 = \{j 3 \leq j \leq n, \text{ and } j-2 \text{ is a multiplication of } 3\},$ $J_3 = \{j 3 \leq j \leq n, \text{ and } j \text{ is a multiplication of } 3\},$
UF9	$f_1 = 0.5[\max\{0, (1+\epsilon)(1-4(2x_1-1)^2)\} + 2x_1]x_2 + \frac{2}{ J_1 } \sum_{j \in J_1} (x_j - 2x_2 \sin(2\pi x_1 + \frac{j\pi}{n}))^2$ $f_2 = 0.5[\max\{0, (1+\epsilon)(1-4(2x_1-1)^2)\} + 2x_1]x_2 + \frac{2}{ J_2 } \sum_{j \in J_2} (x_j - 2x_2 \sin(2\pi x_1 + \frac{j\pi}{n}))^2$ $f_3 = 1 - x_2 + \frac{2}{ J_3 } \sum_{j \in J_3} (x_j - 2x_2 \sin(2\pi x_1 + \frac{j\pi}{n}))^2$ $J_1 = \{j 3 \leq j \leq n, \text{ and } j-1 \text{ is a multiplication of } 3\}, J_2 = \{j 3 \leq j \leq n, \text{ and } j-2 \text{ is a multiplication of } 3\},$ $J_3 = \{j 3 \leq j \leq n, \text{ and } j \text{ is a multiplication of } 3\}, \epsilon = 0.1$
UF10	$f_1 = \cos(0.5x_1\pi) \cos(0.5x_2\pi) + \frac{2}{ J_1 } \sum_{j \in J_1} [4y_j^2 - \cos(8\pi y_j) + 1]$ $f_2 = \cos(0.5x_1\pi) \sin(0.5x_2\pi) + \frac{2}{ J_2 } \sum_{j \in J_2} [4y_j^2 - \cos(8\pi y_j) + 1]$ $f_3 = \sin(0.5x_1\pi) + \frac{2}{ J_3 } \sum_{j \in J_3} [4y_j^2 - \cos(8\pi y_j) + 1]$ $J_1 = \{j 3 \leq j \leq n, \text{ and } j-1 \text{ is a multiplication of } 3\}, J_2 = \{j 3 \leq j \leq n, \text{ and } j-2 \text{ is a multiplication of } 3\},$ $J_3 = \{j 3 \leq j \leq n, \text{ and } j \text{ is a multiplication of } 3\},$

表 3 UF1 至 UF10 的 IGD 统计结果

IGD	UF1 (bi-objective)			UF2 (bi-objective)			UF3 (bi-objective)		
	MOGWO	MOPSO	MOEA/D	MOGWO	MOPSO	MOEA/D	MOGWO	MOPSO	MOEA/D
Average	0.114425	0.137005	0.187135	0.05825	0.060405	0.122343	0.255691	0.313999	0.288648
Median	0.113	0.131745	0.182855	0.057775	0.04835	0.120135	0.25091	0.30802	0.289295
STD. Dev.	0.019538	0.044068	0.05073	0.007392	0.027625	0.0107	0.080702	0.044726	0.015921
Worst	0.15774	0.22786	0.24642	0.07322	0.13051	0.14369	0.36786	0.37773	0.31294
Best	0.08023	0.0899	0.12652	0.0498	0.03699	0.10486	0.1295	0.25648	0.26342
IGD	UF4 (bi-objective)			UF5 (bi-objective)			UF6 (bi-objective)		
	MOGWO	MOPSO	MOEA/D	MOGWO	MOPSO	MOEA/D	MOGWO	MOPSO	MOEA/D
Average	0.058669	0.136037	0.068131	0.797072	2.202376	1.291451	0.279375	0.64752	0.688119
Median	0.058685	0.13432	0.06846	0.69942	2.125745	1.33761	0.24435	0.55073	0.698415
STD. Dev.	0.000481	0.007391	0.002143	0.378579	0.553042	0.134897	0.104485	0.266117	0.055326
Worst	0.05936	0.15189	0.07037	1.73857	3.03836	1.46746	0.55036	1.24281	0.74011
Best	0.05797	0.12733	0.06466	0.46795	1.46479	1.12306	0.19338	0.37933	0.55235
IGD	UF7 (bi-objective)			UF8 (tri-objective)			UF9 (tri-objective)		
	MOGWO	MOPSO	MOEA/D	MOGWO	MOPSO	MOGWO	MOPSO	MOGWO	MOPSO
Average	0.160359	0.353954	0.455242	2.057772	0.536709	0.191747	0.488503	3.594533	1.637196
Median	0.07336	0.387305	0.437665	2.335965	0.5364	0.166025	0.41451	2.82552	1.59163
STD. Dev.	0.139111	0.204421	0.189831	1.145524	0.182571	0.092504	0.144497	3.488293	0.298794
Worst	0.40142	0.61512	0.67701	3.87888	0.79637	0.44794	0.7221	12.95643	2.1622
Best	0.06275	0.05402	0.029	0.46131	0.2453	0.1291	0.33355	1.0431x4	1.22008

表 4 UF1 至 UF10 的 SP 统计结果

IGD	UF1 (bi-objective)			UF2 (bi-objective)			UF3 (bi-objective)		
	MOGWO	MOPSO	MOEA/D	MOGWO	MOPSO	MOEA/D	MOGWO	MOPSO	MOEA/D
Average	0.01237	0.00898	0.00384	0.01108	0.00829	0.00876	0.04590	0.00699	0.02680
Median	0.00536	0.00855	0.00382	0.00946	0.00814	0.00859	0.04860	0.00677	0.02505
STD. Dev.	0.01462	0.00247	0.00151	0.00362	0.00168	0.00076	0.01453	0.00170	0.02064
Worst	0.04641	0.01464	0.00665	0.01816	0.01245	0.01042	0.07050	0.01007	0.06256
Best	0.00081	0.00670	0.00213	0.00758	0.00624	0.00797	0.01549	0.00476	0.00078
IGD	UF4 (bi-objective)			UF5 (bi-objective)			UF6 (bi-objective)		
	MOGWO	MOPSO	MOEA/D	MOGWO	MOPSO	MOEA/D	MOGWO	MOPSO	MOEA/D
Average	0.00969	0.00666	0.00730	0.15231	0.00479	0.00278	0.01446	0.02084	0.00630
Median	0.00857	0.00662	0.00728	0.08778	0.00487	0.00007	0.01110	0.01235	0.00000
STD. Dev.	0.00390	0.00091	0.00059	0.16247	0.00408	0.00553	0.01246	0.03258	0.01267
Worst	0.01722	0.00809	0.00836	0.51247	0.01206	0.01615	0.04112	0.11140	0.03030
Best	0.00583	0.00546	0.00610	0.00843	0.00006	0.00000	0.00191	0.00215	0.00000
IGD	UF7 (bi-objective)			UF8 (tri-objective)			UF9 (tri-objective)		
	MOGWO	MOPSO	MOEA/D	MOGWO	MOPSO	MOGWO	MOPSO	MOGWO	MOPSO
Average	0.00824	0.00670	0.00540	0.00687	0.02682	0.01743	0.02343	0.02523	0.01994
Median	0.00548	0.00655	0.00441	0.00470	0.02639	0.01826	0.02349	0.02392	0.02066
STD. Dev.	0.00856	0.00285	0.00301	0.00474	0.00827	0.00633	0.00405	0.01500	0.00348
Worst	0.03106	0.01240	0.01168	0.01879	0.04473	0.02856	0.03087	0.05384	0.02665
Best	0.00031	0.00325	0.00084	0.00365	0.01531	0.00653	0.01716	0.00000	0.01536

表 5 UF1 至 UF10 的 MS 统计结果

IGD	UF1 (bi-objective)			UF2 (bi-objective)			UF3 (bi-objective)		
	MOGWO	MOPSO	MOEA/D	MOGWO	MOPSO	MOEA/D	MOGWO	MOPSO	MOEA/D
Average	0.92680	0.64538	0.51774	0.90972	0.91205	0.87201	0.94982	0.61030	0.23994
Median	0.93272	0.66322	0.59541	0.91041	0.91636	0.87437	1.00000	0.61612	0.22943
STD. Dev.	0.06884	0.19292	0.16609	0.02867	0.02560	0.00560	0.08777	0.10575	0.12129
Worst	0.81797	0.26592	0.31487	0.84695	0.86654	0.85986	0.76809	0.38172	0.08975
Best	0.99711	0.95226	0.74128	0.94791	0.95301	0.87794	1.00000	0.77145	0.47863
IGD	UF4 (bi-objective)			UF5 (bi-objective)			UF6 (bi-objective)		
	MOGWO	MOPSO	MOEA/D	MOGWO	MOPSO	MOEA/D	MOGWO	MOPSO	MOEA/D
Average	0.94242	0.81275	0.88320	0.39503	0.27926	0.29215	0.67360	0.27435	0.09677
Median	0.94269	0.81321	0.88131	0.43258	0.28654	0.29165	0.70826	0.22917	0.00005
STD. Dev.	0.00093	0.01367	0.01812	0.17494	0.09575	0.03470	0.12323	0.11285	0.20715
Worst	0.94095	0.79441	0.85324	0.03006	0.15574	0.23834	0.38838	0.15436	0.00000
Best	0.94327	0.83449	0.91394	0.61042	0.43827	0.34380	0.81492	0.52516	0.59484
IGD	UF7 (bi-objective)			UF8 (tri-objective)			UF9 (tri-objective)		
	MOGWO	MOPSO	MOEA/D	MOGWO	MOPSO	MOGWO	MOPSO	MOGWO	MOPSO
Average	0.80126	0.42928	0.56317	0.44573	0.50810	0.83991	0.19816	0.29721	0.13015
Median	0.96293	0.29520	0.63267	0.44429	0.50601	0.91055	0.16566	0.14238	0.10913
STD. Dev.	0.30865	0.27553	0.24209	0.18574	0.16136	0.19759	0.16351	0.34651	0.06263
Worst	0.02252	0.14458	0.14963	0.18863	0.22723	0.28750	0.06771	0.03194	0.06489
Best	0.98746	0.87714	0.99152	0.86376	0.71476	0.93753	0.64242	0.92828	0.25404