

NSGA-II 算法

姜孟冯^{1,2}

(1. 中国矿业大学, 徐州; 2. 应急管理部信息研究院, 北京)

日期: 2024 年 10 月 30 日

1 算法背景

1.1 多目标优化问题

多目标优化问题 (MOP) 是涉及多个目标函数同时优化的数学问题。需要在两个或多个相互冲突的目标之间进行权衡的情况下作出最优决策。

定义 1.1. 多目标优化问题 (multi-objective optimization problem, 简称 MOP)

$$\begin{aligned} \min \quad & y = (f_1(x), f_2(x), \dots, f_r(x)) \\ \text{s.t.} \quad & g_i(x) \geq 0, i = 1, 2, \dots, k, \\ & h_i(x) = 0, i = 1, 2, \dots, l \end{aligned}$$

这里 $x = (x_1, x_2, \dots, x_n)$ 是一个 n 维决策向量, 向量空间 $X \subseteq \mathbf{R}^n$, y 是一个 r 维目标向量 $y \in \mathbf{R}^r$, f 为目标函数, g 和 h 分别为不等式约束与等式约束条件, 记问题的可行解集为 Ω 。

定义 1.2. Pareto 支配 (Pareto Dominance)

设 $a, b \in \Omega$, a 支配 b , 记为 $a \prec b$, 当且仅当以下条件同时成立

$$\forall i \in \{1, 2, \dots, r\}, f_i(a) \leq f_i(b) \text{ 且 } \exists j \in \{1, 2, \dots, r\}, f_j(a) < f_j(b)$$

定义 1.3. Pareto 最优解集 (Pareto Optimal Set)

如果一个可行解 $x^* \in \Omega$ 被称之为 Pareto optimal solution, 当且仅当 x 不被其他的解支配。易见 x^* 往往不是一个, 用 $PS^* = \{x^*\}$ 表示最优解集

$$PS^* = \{x^*\} = \{x \in \Omega | \neg \exists x' \in \Omega, f_j(x') \leq f_j(x) (j = 1, 2, \dots, r)\}$$

定义 1.4. Pareto 边界 (Pareto Optimal Front, 简称 PF)

Pareto Optimal Set 中每个解对应的目标值向量组成的集合称之为 Pareto Front (如图 1 所示)。

$$PF^* = \{f(x) = (f_1(x), f_2(x), \dots, f_r(x)) | x \in PS^*\}$$

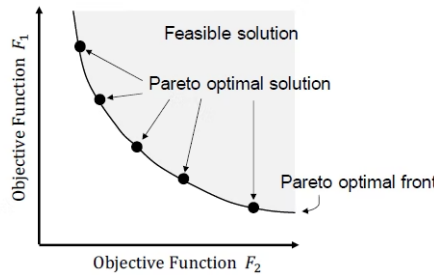


图 1: Pareto 边界示意图

1.2 多目标进化算法与分类

多目标进化算法 (Multi-objective Evolutionary Algorithm, 简称 MOEA) 是一种非常有效的解决多目标优化问题 (MOP) 的方法。1989 年, David Goldberg 在其著作《Genetic Algorithm in Search, Optimization and Machine Learning^[1]》中, 提出了用进化算法实现多目标的优化技术, 对多目标进化算法的研究具有重要的方向性指导意义。根据不同的进化机制, MOEA 可以分为三类^[2]:

1. 基于分解的 MOEA: 权重聚合函数、切比雪夫聚合、基于惩罚的边界交叉、MOEA/D
2. 基于支配关系的 MOEA: NSGA-II、NPGA、SPEA2、PESA、PAES、MOMGA、mBOA、MMOGA
3. 基于指标的 MOEA: SMS-EMOA、IBEA、HypE

1.3 NSGA-II 算法

1994 年, Srinivas 和 Deb 提出了 NSGA(nondominated sorting genetic algorithm)^[3], 可以有效求解 MOP 问题, 但仍存在一些不足之处: (1) 缺少最优个体保留机制 (2) 共享参数大小不易确定 (3) 构造非支配集的时间复杂度高 ($O(rN^3)$)。2000 年, Deb 等对算法进行优化, 提出了 NSGA-II^[4], 2002 年又对其中的非支配集排序方法进行了改进^[5], 把时间复杂度提高到 ($O(rN^2)$), 但只适用于处理低维优化问题 (维数小于等于 3)。2013 年, Deb 等又提出了 NSGA-III^[6], 用以解决高维数问题。

本文主要集中讨论 NSGA-II 算法。

2 NSGA-II 算法原理及框架

2.1 算法框架

NSGA-II 算法是一种多目标进化算法, 全称是 Elitist Non-dominated Sorting Genetic Algorithm (精英非支配排序遗传算法), NSGA-II 的主要特征有三点:

- 基于快速非支配排序的等级 (rank) 划分方法
- 计算拥挤度 (crowding-distance) 的方法
- 用以上二者构建的偏序比较算子代替共享参数 (sharing function)

NSGA-II 算法基于遗传算法的迭代进化思想, 开始时随机产生一个父代种群 P_t , 然后利用选择、交叉、变异的方式生成临时子代 Q_t , 利用“非支配排序”与“拥挤度计算”来对父子种群中的每个个体进行适应度评估, 选出新一代精英个体作为新的父代 P_{t+1} , 如此不断循环进化, 直到满足结束条件。

NSGA-II 基本流程框架如图 2 所示:

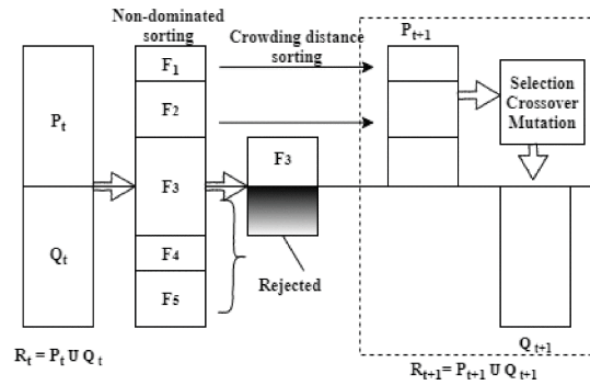


图 2: NSGA-II 流程示意图

2.2 快速非支配集排序方法

在 NSGA 中, 非支配集排序方法的时间复杂度是 $(O(rN^3))$, NSGA-II 改进了非支配集的构造方式。设向量 S_p 表示个体 p 所支配的个体集合, 标量 n_p 表示能够支配当前个体 p 的个数, 即有

$$S_p = \{q | p \prec q, p, q \in \text{种群} P\}, \quad n_p = |\{q | q \prec p, p, q \in \text{种群} P\}|$$

可利用上述变量及个体间的支配关系对种群 P 进行划分, 输出 $F_i (i = 1, 2, \dots, m)$.

$$F_i = \{q | n_q - i + 1 = 0, q \in P\}$$

```

fast-non-dominated-sort( $P$ )
for each  $p \in P$ 
     $S_p = \emptyset$ 
     $n_p = 0$ 
    for each  $q \in P$ 
        if  $(p \prec q)$  then
             $S_p = S_p \cup \{q\}$ 
             $n_p = n_p + 1$ 
        else if  $(q \prec p)$  then
             $n_p = n_p + 1$ 
    if  $n_p = 0$  then
         $p_{\text{rank}} = 1$ 
         $\mathcal{F}_1 = \mathcal{F}_1 \cup \{p\}$ 
    if  $p$  dominates  $q$ 
        Add  $q$  to the set of solutions dominated by  $p$ 
    Increment the domination counter of  $p$ 
     $p$  belongs to the first front

 $i = 1$ 
while  $\mathcal{F}_i \neq \emptyset$ 
     $Q = \emptyset$ 
    for each  $p \in \mathcal{F}_i$ 
        for each  $q \in S_p$ 
             $n_q = n_q - 1$ 
            if  $n_q = 0$  then
                 $q_{\text{rank}} = i + 1$ 
                 $Q = Q \cup \{q\}$ 
             $q$  belongs to the next front
     $i = i + 1$ 
     $\mathcal{F}_i = Q$ 
    Initialize the front counter
    Used to store the members of the next front

```

算法 fast-non-dominated-sort 由两部分组成, 第一部分用于计算 S_p 与 n_p 并构造 F_1 , 种群规模为 N , 目标数为 r 时, 时间复杂度为 $O(rN^2)$ 。第二部分用于求解 F_i , 最坏情况下存在 N 层边界, 即 $m=N$, 时间复杂度为 $O(N^2)$ 。由此可知, 算法的时间复杂度为 $O(rN^2) + O(N^2) = O(rN^2)$ 。

2.3 拥挤度计算方法

设下标为 i 的个体的拥挤度记为 i_{distance} , 目标函数值记为 $f_j(x_i)$, 用 f_j^{\max} 与 f_j^{\min} 分别表示目标函数的最大最小值, 则拥挤度计算可使用以下公式:

$$\begin{aligned} \text{边界上的点: } i_{\text{distance}} &= \infty \\ \text{边界以外的点: } i_{\text{distance}} &= \sum_{j=1}^r \left(\frac{f_j(x_{i+1}) - f_j(x_{i-1})}{f_j^{\max} - f_j^{\min}} \right) \end{aligned}$$

```

crowding-distance-assignment( $\mathcal{I}$ )
 $l = |\mathcal{I}|$ 
for each  $i$ , set  $\mathcal{I}[i]_{\text{distance}} = 0$ 
for each objective  $m$ 
     $\mathcal{I} = \text{sort}(\mathcal{I}, m)$ 
     $\mathcal{I}[1]_{\text{distance}} = \mathcal{I}[l]_{\text{distance}} = \infty$ 
    for  $i = 2$  to  $(l - 1)$ 
         $\mathcal{I}[i]_{\text{distance}} = \mathcal{I}[i]_{\text{distance}} + (\mathcal{I}[i + 1].m - \mathcal{I}[i - 1].m) / (f_m^{\max} - f_m^{\min})$ 
    number of solutions in  $\mathcal{I}$ 
    initialize distance
    sort using each objective value
    so that boundary points are always selected
    for all other points

```

算法 `crowding-distance-assignment` 描述了计算个体拥挤度的过程，此过程需要按每个子目标函数值进行排序，最坏情况下对 r 个子目标进行排序，时间复杂度为 $O(rN \log N)$ ，而计算拥挤度本身的时间为 $O(rN)$ ，由此可知算法的时间复杂度为 $O(rN \log N)$ 。

2.4 精英保留策略

NSGA-II 的适应度评估过程中，考虑了个体 i 的两个属性，非支配排序等级与拥挤度，记作 i_{rank} 与 $i_{distance}$ ，二者可以构成一个偏序关系 \prec_n 。

定义 2.1. 个体之间的偏序关系 \prec_n

$$i \prec_n j \quad \text{if } (i_{rank} < j_{rank}) \quad \text{or} \quad ((i_{rank} = j_{rank}) \quad \text{and} \quad (i_{distance} > j_{distance}))$$

利用该偏序关系，可以构建偏序比较算子，该算子避免了任何用户定义参数，同时改善了计算复杂度。如此可定义精英保留策略如下：

1. 将父代种群 P_t 通过选择、交叉、变异策略生成子代种群 Q_t
2. 把 P_t 和 Q_t 合成种群 R_t
3. 根据偏序关系 \prec_n 将种群 R_t 中的个体逐个放入新的父代种群 P_{t+1} ，直到填满。

3 算法步骤及实现

3.1 基本步骤

NSGA-II 主循环算法可用伪代码描述如下：

$R_t = P_t \cup Q_t$ $\mathcal{F} = \text{fast-non-dominated-sort}(R_t)$ $P_{t+1} = \emptyset$ and $i = 1$ until $ P_{t+1} + \mathcal{F}_i \leq N$ $\text{crowding-distance-assignment}(\mathcal{F}_i)$ $P_{t+1} = P_{t+1} \cup \mathcal{F}_i$ $i = i + 1$ $\text{Sort}(\mathcal{F}_i, \prec_n)$ $P_{t+1} = P_{t+1} \cup \mathcal{F}_i[1 : (N - P_{t+1})]$ $Q_{t+1} = \text{make-new-pop}(P_{t+1})$ $t = t + 1$	combine parent and offspring population $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, \dots)$, all nondominated fronts of R_t until the parent population is filled calculate crowding-distance in \mathcal{F}_i include i th nondominated front in the parent pop check the next front for inclusion sort in descending order using \prec_n choose the first $(N - P_{t+1})$ elements of \mathcal{F}_i use selection, crossover and mutation to create a new population Q_{t+1} increment the generation counter
--	--

具体来说，NSGA-II 使用快速非支配排序来保证收敛性，并且利用拥挤距离来保证分布性。其主要时间开销由三部分组成：

1. 构造非支配集划分 (fast-non-dominated-sort): $O(r(2N)^2)$
2. 拥挤度计算 (crowding-distance-assignment): $O(r(2N) \log(2N))$
3. 构造偏序集 (sort on \prec_n): $O(2N \log 2N)$

综合考虑，算法的时间复杂度为： $O(rN^2)$ 。

3.2 参数设置

NSGA-II 算法的参数设置对算法的性能有重要影响。主要参数包括：

- 种群规模 (N)：控制种群中个体的数量。较大的种群规模可以提供更广泛的搜索空间，但计算成本更高。

- 最大进化代数 (MaxGen): 控制算法运行的代数。较大的最大进化代数可以提高算法的收敛性, 但计算成本也更高。
- 交叉概率 (Pc): 控制交叉算子应用的概率。较高的交叉概率可以促进种群多样性, 但可能导致过早收敛。
- 变异概率 (Pm): 控制变异算子应用的概率。较高的变异概率可以引入新的个体, 但可能破坏种群的收敛性。

3.3 开源实现

C nsga2: <https://www.iitk.ac.in/kangal/codes.shtml>

Java MOEA Framework: <https://github.com/MOEAFramework/MOEAFramework>

Python pymoo: <https://github.com/anyoptimization/pymoo>

Julia Metaheuristics: <https://github.com/jmeja8/Metaheuristics.jl>

R rmoo: <https://github.com/Evolutionary-Optimization-Laboratory/rmoo>

Rust Optirustic: <https://github.com/s-simoncelli/optirustic>

C# MOEAs: <https://github.com/qshzhang/MOEAs>

可见, NSGA-II 算法作为多目标优化 (MOO) 领域的经典算法, 被众多编程语言予以实现, 方便后人研究与改进。

4 算法应用

NSGA-II 算法应用非常广泛,

4.1 科学问题研究

Shanu Verma 等在 2021 年对 NSGA-II 相关的论文发表情况进行了分析^[7], 从 Science Direct、Taylor & Francis Online、Wiley Online Library、Springer 与 IEEE Xplore 等在线数据库上选出了 169 篇 NSGA-II 论文。按照多目标组合优化问题的各类子问题进行统计, 得出结果如图 3 所示。调度问题 (占比 42%) 是最适用 NSGA-II 算法最解决的多目标优化问题 (MOCOP) 中最为常见的, 紧随其后的是分配问题 (占比 17%)、其他问题 (占比 11%)、指派问题 (占比 11%)、车辆路径问题 (VRP) (占比 8%)、背包问题 (占比 7%) 以及旅行商问题 (TSP) (占比 4%)。在应用算法方面, 改进后的 NSGA-II 算法 (占比 58%) 被最广泛地使用, 其次是混合 NSGA-II 算法 (占比 34%), 而传统的 NSGA-II 算法使用较少, 仅占 8%。

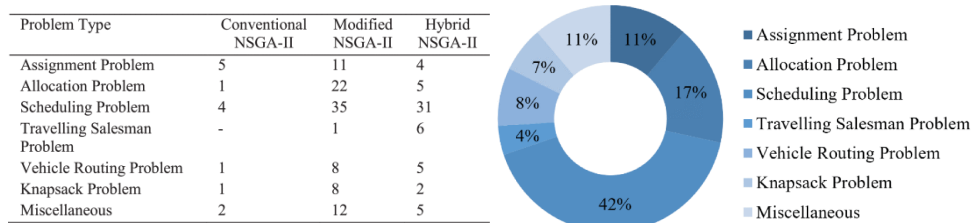


图 3: NSGA-II 研究论文按问题分类

若以年为维度, 如图 4 所示, 可见一直至 2020 年, NSGA-II 的研究热度持续上涨。

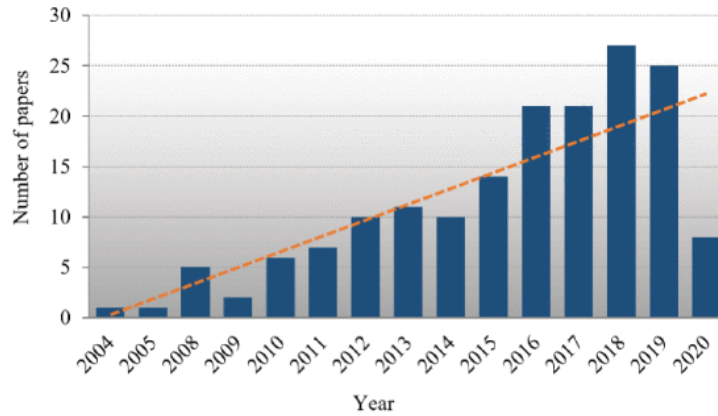


图 4: NSGA-II 研究论文按年的研究热度

4.2 机械设计与制造

机械设计与制造是 NSGA-II 算法最常见的应用领域之一。在机械设计中，往往需要同时考虑多个目标，如成本、性能、可靠性等。NSGA-II 算法可以有效地处理多目标优化问题，找到满足所有目标约束的最佳解决方案。使用 NSGA-II 算法的案例非常之多，比如：车用行星齿轮成形工艺^[8]、缠绕过程多目标工艺^[9]、杯状纵磁触头结构^[10] 等等。

4.3 环境与资源分配

环境与资源分配优化是 NSGA-II 算法的另一个重要应用领域。在资源分配问题中，需要在有限的资源约束下，将资源分配给多个目标，这正是经典的 MOP 问题，比如供水系统规划^{[11][12]}、供电系统规划^{[13][14]}、土地资源利用规划^{[15][16]}、充电站选址^[17] 等，都可以用 NSGA-II 算法有效地解决。

4.4 电子与电气工程

在电子与电气工程方面，NSGA-II 的应用也比较广泛。随着硅片上集成的晶体管数量迅猛增加，基于 IP 核的片上系统 (system on a chip, SOC) 已成为 VLSI 实现技术发展的趋势。目前 SOC 的系统设计主要采用基于可编程 IP 核的配置、仿真与封测，所有功能验证完毕后才产生物理芯片。由于 IP 核的多样性及其可优化参数的矛盾性，使得 SOC 的设计空间极其复杂。SOC 系统的主要任务之一就是针对具体的应用在可能的设计空间中找到一组满足设计约束的 IP 可行配置集，或者是找到功率/性能平衡面，把功耗和执行时间的最小化作为优化目标。案例如：监测天线部署优化^[18]、阵列芯片的微通道结构参数优化^[19]、三模冗余软错误防护技术^[20] 等。

参考文献

- [1] GOLDBERG D E. Genetic Algorithm in Search, Optimization and Machine Learning, Addison[J]. Wesley Publishing Company, Reading, MA, 1989, 1(98): 9.
- [2] 郑金华, 邹娟. 多目标进化优化[M]. 科学出版社, 2017.
- [3] SRINIVAS N, DEB K. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms [J/OL]. Evolutionary Computation, 1994, 2(3): 221-248. DOI: 10.1162/evco.1994.2.3.221.

- [4] DEB K, AGRAWAL S, PRATAP A, et al. A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II[C]//SCHOENAUER M, DEB K, RUDOLPH G, et al. Parallel Problem Solving from Nature PPSN VI. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000: 849-858.
- [5] DEB K, PRATAP A, AGARWAL S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J/OL]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197. DOI: [10.1109/4235.996017](#).
- [6] DEB K, JAIN H. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints[J/OL]. IEEE Transactions on Evolutionary Computation, 2014, 18(4): 577-601. DOI: [10.1109/TEVC.2013.2281535](#).
- [7] VERMA S, PANT M, SNASEL V. A Comprehensive Review on NSGA-II for Multi-Objective Combinatorial Optimization Problems[J/OL]. IEEE Access, 2021, 9: 57757-57791. DOI: [10.1109/ACCESS.2021.3070634](#).
- [8] 吴玥, 龚红英, 尤晋, 等. 基于响应面法和 NSGA-II 的车用行星齿轮成形工艺[J/OL]. 锻压技术, 2024, 49(10): 65-74. DOI: [10.13330/j.issn.1000-3940.2024.10.009](#).
- [9] 韩宇泽, 刘雁鹏, 任中杰, 等. 基于 NSGA-II 算法的缠绕过程多目标工艺参数优化[J/OL]. 复合材料学报, 2024, 41(10): 5622-5633. DOI: [10.13801/j.cnki.fhclxb.20231218.005](#).
- [10] 丁璨, 李江, 袁召, 等. 基于 NSGA-II 和 BP 神经网络的杯状纵磁触头结构优化设计[J/OL]. 电工技术学报, 2022, 37(23): 6074-6082. DOI: [10.19595/j.cnki.1000-6753.tces.220369](#).
- [11] 苏彦博, 吕谋, 褚洋洋. 基于 NSGA- 的供水管网漏损定位多目标优化模型[J/OL]. 水电能源科学, 2023, 41(07): 132-136. DOI: [10.20040/j.cnki.1000-7709.2023.20221893](#).
- [12] 谭昭, 陈红勋, 马峥, 等. 基于多目标遗传算法 (NSGA-) 的供水泵站运行优化研究[J/OL]. 水电能源科学, 2022, 40(12): 181-185. DOI: [10.20040/j.cnki.1000-7709.2022.20220421](#).
- [13] 王艳, 张嘉琳, 赵洪山. 基于 Tent 映射 NSGA- 算法的微电网多目标优化方法[J]. 科学技术与工程, 2022, 22(14): 5643-5652.
- [14] 张会林, 欧阳婷, 杨海马. 基于自适应混沌改进 NSGA- 的微网需求响应配置[J/OL]. 控制工程, 2023, 30(01): 47-53. DOI: [10.14107/j.cnki.kzgc.20210055](#).
- [15] 赵丞豪, 张学霞, 姜东旸, 等. 基于 NSGA-II 算法的若尔盖高原高寒湿地生态修复格局优化[J/OL]. 中国水土保持科学 (中英文), 2024, 22(02): 55-64. DOI: [10.16843/j.sswc.2022021](#).
- [16] 韩珊, 王潜心, 曹芳洁, 等. 多目标土地利用优化模型与应用[J/OL]. 测绘科学, 2022, 47(08): 157-165. DOI: [10.16251/j.cnki.1009-2307.2022.08.018](#).
- [17] 俞宁, 冯鑫, 汤爱华, 等. 基于改进 NSGA-II 算法的电动汽车充电站选址方法[J/OL]. 郑州大学学报 (理学版), 2024: 1-6. DOI: [10.13705/j.issn.1671-6841.2024133](#).
- [18] 杜文占, 余志勇, 杨剑, 等. 改进 NSGA-II 算法及监测天线部署优化研究[J/OL]. 西安电子科技大学学报, 2021, 48(05): 239-248. DOI: [10.19665/j.issn1001-2400.2021.05.027](#).
- [19] 王康宁, 陈宸, 杜加友, 等. 基于 NSGA-II 算法的微通道结构参数优化[J/OL]. 杭州电子科技大学学报 (自然科学版), 2024, 44(04): 94-102. DOI: [10.13954/j.cnki.hdu.2024.04.012](#).
- [20] 李炎, 胡岳鸣, 曾晓洋. 面向商业航天卫星成本效益的三模冗余软错误防护技术: 近似计算的实践[J]. 电子与信息学报, 2024, 46(05): 1604-1612.