

一种快速、精英的多目标遗传算法：NSGA-II

Kalyanmoy Deb

副成员：IEEE, Amrit Pratap, Sameer Agarwal 和 T.Meyarivan

摘要——抽象多目标进化算法（EAs）使用非支配的排序和共享的行为很重要

- 1) 计算复杂性（目标数量和人口大小）；
- 2) 非政治方法；
- 3) 需要指定共享参数。

在本文中，我们建议基于排序的多目标 EA（MOEA），称为非支配的排序遗传算法 II（NSGA-II），它减轻了以上三个困难。

特别是，快速非支配具有计算复杂性的排序方法被提出了。此外，还提供了一个选择运算符，用于创建结合亲子群体的交配池，选择最好的（关于健身和传播）解决方法。对测试难题的仿真结果表明在大多数问题中，拟议的 NSGA-II 能够找到更好的解觉传播和更接近真实的收敛性 pareto 最优前沿与 pareto 存档进化战略和实力帕累托是另外两个精英特别注意创造多样化的帕累托最优前沿。

此外，我们修改了支配地位的定义，以便有效地解决约束多目标问题。模拟一些测试问题的约束 NSGA-II 的结果，包括一个五目标七约束非线性问题与另一个约束多目标优化器相比观察到 nsga-ii 具有更好的性能。索引项约束处理，精英主义，遗传算法，多准则决策，多目标优化，帕累托最优解。

一、导言

一个问题中存在多个目标原理，产生一组最优解（很大程度上称为帕累托最优解），而不是单个最优解解决方案。在没有进一步信息的情况下帕累托最优解不能说比其他。这就要求用户找到尽可能多的 pareto 最优解。经典优化方法（包括多准则决策方法）建议转换强调一个帕累托最优的单目标优化问题的多目标优化问题一次解决。当这种方法被用来寻找多个解决方案，必须多次应用，希望如此在每次模拟运行时查找不同的解决方案。在过去的十年里，人们提出了许多多目标进化算法[1]、[7]、[13]，2000 年 8 月 18 日收到的手稿；2001 年 2 月 5 日修订 2001 年 9 月 7 日。K. Deb 的工作得到了该部的支持印度人力资源与发展部发展计划。作者来自印度坎普尔理工学院坎普尔遗传算法实验室，坎普尔 pin 208 016，印

度（电子邮件：deb@iitk.ac.in）。发布者项目标识符 S 1089-778X (02) 04101-2。[20]，[26]。主要原因是他们有能力一次模拟运行中的多重帕累托最优解。因为进化算法（EAS）适用于解决方案，一个简单的 ea 可以扩展到一套解决方案。强调走向真实 pareto 最优区域，一个 ea 可以用来找到多个一次模拟运行的 pareto 最优解。文[20]中提出的非支配排序遗传算法（nsga）是最早提出的一种非支配排序遗传算法。多年来，对 nsga 方法的主要批评如下。1）非支配排序计算复杂度高：当前使用的非支配排序算法有计算复杂度目标数量和人口规模）。这个使得 nsga 在计算上对于较大的种群规模是昂贵的。这种巨大的复杂性是由于每一代非支配排序过程的复杂性。

2）缺乏精英主义：最近的研究结果[25]，[18]表明精英主义可以显著提高遗传算法的性能，这也有助于防止失去好的解决方案一旦他们被发现。3）需要指定共享参数：确保人口多样性的传统机制为了得到各种各样的等效解主要是关于分享的概念。主要问题是共享是指它需要指定共享参数（ σ ）。尽管有一些工作共享参数的动态大小调整[10]，需要一种无参数多样性保持机制。在本文中，我们将讨论所有这些问题，并提出改进版的 nsga，我们称之为 nsga-ii。从在一些困难的测试问题上，我们发现 NSGA-II 优于其他两个当代 MOEA：pareto 存档进化策略（paes）[14]和 strengthpareto-ea（spea）[24]在寻找一组不同的解和在接近真实 pareto 最优集的收敛方面。约束多目标优化在解决实际问题的观点，但不太重视到目前为止，ea 研究人员在这方面已经获得了报酬。本文提出了一种简单的约束处理策略与 NSGA-II 一起，非常适合任何 EA。关于四个问题选自文献，NSGA-II 与另一个最近提出的约束处理策略。这些结果鼓励将 NSGA-II 应用于更复杂的以及现实世界中的多目标优化问题。在本文的其余部分，我们简要地提到第二节现存的精英分子。此后，在第三节中，详细描述了所提出的 nsga-ii 算法。第四节给出了 nsga-ii 的仿真结果并进行了比较。他们和另外两个精英莫亚斯（paes 和 spea）。在第五节中，我们强调了参数相互作用的问题这在进化计算研究中是很重要的。下一个节扩展 NSGA-II 以处理约束并比较另一个最近提出的约束处理的结果方法。最后，我们概述了本文的结论。

二、精英多目标进化算法

在 1993-1995 年间，人们提出了一些不同的算法来解决多目标优化问题。其中之一，Fonseca 和 Fleming 的 Moga[7]，Srinivas 和 Deb 的 NSGA[20]和 Horn 等人的 NPGA[13]受到了更多的关注。这些算法证明了将简单的 ea 转换为 moea 所需的附加算子。两个普通的这三个操作符的特性如下：

- i) 赋值基于非支配排序的群体成员适应度

ii) 保持相同解决方案之间的多样性非支配阵线。尽管他们发现多个测试问题的多重非支配解许多工程设计问题, 研究人员意识到需要引入更多有用的运算符在单目标环境分析中(被发现是有用的), 以便解决多目标优化问题更好。尤其, 我们的兴趣是引进精英主义来加强 moea 的收敛性。参考文献[25]显示精英主义有助于在教育部实现更好的融合。

在现有的精英莫亚斯, 齐兹勒和蒂勒的 spea 中[26], Knowles 和 Corne 的 Pareto 存档了 PAES[14], 以及鲁道夫的精英 GA[18]研究得很好。我们描述这些简而言之。详情请参阅对最初的研究。Zitzler 和 Thiele[26]提出了一个精英的多准则环境在他们的 spea 中有非支配的概念。他们建议每一代人都要保持外来人口存储迄今为止发现的所有非支配解决方案从最初的人口。这个外部群体参与了所有的遗传操作。在每一代人中, 外来人口和流动人口居首位构建。组合种群中的所有非支配解都根据解的个数赋予一个适应度。它们支配和支配的解决方案被分配适应度比任何非支配解的最坏适应度都差。这个适应度分配确保搜索是定向的走向非支配的解决方案。确定性聚类技术用于确保非支配解决。尽管[26]中建议的实现是在适当的簿记下 spea 的复杂性可以简化为。knowles 和 corne[14]建议使用一个简单的 moea 类似于(11)进化的单亲单亲战略。二进制字符串不是使用实数参数, 而是使用和按位突变创造后代。在他们的 PAE 中, 有一个父母和一个后代与父母相比。如果子代支配父代, 则子代被接受为下一个父代, 并且迭代将继续。另一方面, 如果父母控制了后代, 后代就会被丢弃, 并找到一个新的突变解决方案(新的后代)。然而, 如果后代和父母不相互支配, 则通过比较后代和父母来做出选择拥有迄今为止找到的最佳解决方案的档案。后代是与存档进行比较以检查它是否支配任何成员档案馆的。如果是的话, 后代就被认为是新的父解决方案和所有主导解决方案都从档案文件。如果后代不支配存档, 检查父代和子代是否与存档的解决方案接近。如果后代居住在目标空间中存档成员中最不拥挤的区域, 它被接受为父级并被添加去档案馆。拥挤是通过分割整个在子空间中确定搜索空间, 其中 depth 参数和是决策变量的数目, 以及通过动态更新子空间。调查人员已经计算出了用于评估的 paes 最坏情况的复杂性。因为, 档案的长度在哪里。自从档案馆大小通常与人口大小成比例选择, 算法的总体复杂度为。鲁道夫(rudolph)[18]建议, 但没有模拟, 一个简单的精英主义者 moea 基于系统比较亲代和后代群体。非支配解将子代种群的解与父代种群的解进行比较, 形成一个整体的非支配解集。成为下一次迭代的父总体。如果尺寸此集合中的不大于所需的总体大小, 其他包括后代中的个体。与这个策略, 他证明了这个算法的收敛性帕累托最优前沿。虽然这是一个重要的成就就其本身而言, 该算法缺乏对第二个保持帕累托最优解多样性的任务。必须添加一个显式的多样性保持机制更实用。因为第一个非支配的决定论前面是, 鲁道夫算法

的整体复杂度也是。在下面，我们将提出非支配的采用快速非支配排序的排序遗传算法程序，精英保留方法，无参数刻痕操作员。

三、精英非支配排序遗传算法

A.快速非支配排序法

为了清楚起见，我们首先描述一个天真而缓慢的将种群分为不同的非显性的过程水平。之后，我们将描述一种快速方法。在一个幼稚的方法中，为了确定第一个人口规模中的非支配前沿，每个解可以与人口中的所有其他解决方案进行比较找出它是否被支配。这需要比较每个解决方案，其中是目标数。当这个过程是不断寻找所有成员的第一个非支配水平，总的复杂度是。在这个阶段，第一非支配阵线的所有个人找到了。为了找到下一个非支配的个体前面，第一个前面的解决方案暂时打折重复上述步骤。在最坏的情况下，任务找到第二个前线也需要计算，特别是当解的数目属于第二个和更高的非支配水平。这个论点是对于发现第三和更高层次的非支配性是正确的。因此，最坏的情况是当有锋而只有锋每一个方面都有一个解决方案。这需要一个整体计算。请注意，此过程需要存储。在下面的段落中，方程式显示在在这一页的底部，我们描述了一个快速的非支配排序。需要计算的方法。首先，对于每个解，我们计算两个实体：1) 支配计数，支配解决方案，和 2)，解决方案占主导地位的一组解决方案。这需要比较。第一非支配阵线的所有解决方案统治算作零。现在，对于每个解决方案，我们访问其集合的每个成员（），并将其控制计数减少一。在这样做的时候，如果任何成员的控制计数变为零，我们将其放在一个单独的列表中。这些成员属于第二非支配阵线。现在，上述程序继续执行第三条战线已经确定。这一进程一直持续到各条战线已识别。对于第二级或更高级别的非支配解，支配数最多可以是。因此，每种解决方案在控制计数变为零。在这一点上，解决方案是分配一个非支配级别，以后不再访问。因为最多有这样的解决方案，总的复杂度是。因此，整个过程的复杂性是。计算这种复杂性的另一种方法是认识到第一个内环（每个内环）的主体是执行的时间正好是每个人可以成为成员的时间最多一个前部和第二个内部环路（每个）可在每个个体的最大时间执行[每一个人最大限度地支配着个人，每次控制检查最多需要比较]结果在整个计算中。需要注意的是虽然时间复杂度已经降低到存储需求已增加到。

B.多样性保护

我们之前提到过，以及帕累托最优集，也期望 ea 保持一个好的解在所得到的解集中的扩散。原作 NSGA 使用了众所周知的共享函数方法已经发现，在适当设置相关参数的情况下，可以保持种群的可持续多样性。这个共享函数方法包含一个共享参数，它设置问题所需的共享范围。此参数与选择用于计算两个人口成员之间的接近度。参数表示该距离度量的最大值

其中任何两个解决方案共享彼此的适合性。这个参数通常由用户设置，尽管存在一些指导方针[4]。这种共享函数方法有两个困难。

- 1) 共享函数方法在保持解决方案的分布在很大程度上取决于选择 `?? ??` 的值。

```

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
             $S_p = S_p \cup \{q\}$ 
        else if  $(q \prec p)$  then
             $n_p = n_p + 1$ 
    if  $n_p = 0$  then
         $p_{\text{rank}} = 1$ 
         $\mathcal{F}_1 = \mathcal{F}_1 \cup \{p\}$ 
 $i = 1$ 
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
                 $q_{\text{rank}} = i + 1$ 
                 $Q = Q \cup \{q\}$ 
     $i = i + 1$ 
     $\mathcal{F}_i = Q$ 

```

如果支配
添加到由
否则，如果
增加
如果那属于第一阵线
初始化前计数器
虽然
用于存储下一个前线的成员
为每个人
为每个人
如果那属于下一个前线

deb 等人：一个快速的精英多目标遗传算法：nsga-ii

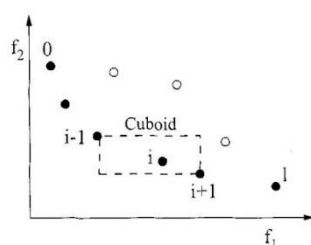


图 1。拥挤距离计算。用实心圆圈标记的点是同一非支配前沿的解。

- 2) 由于每个解决方案都必须与总体中的所有其他解决方案进行比较，因此共享函数方

法是

在建议的 `nsga-ii` 中，我们替换了共享函数使用拥挤的比较方法来消除以上两方面都存在一定的困难。新方法不需要任何用户定义的参数来维护人口成员的多样性。此外，该方法具有更好的计算复杂度。来描述这个方法，我们首先定义一个密度估计度量，然后呈现拥挤的比较运算符。

1) 密度估算：估算围绕人口中某一特定解的解，我们计算两侧两点的平均距离这一点沿着每个目标。这个数量作为长方体周长的估计使用最近的邻居作为顶点(称之为拥挤距离)在图 1 中，在它的正面(用实心圆圈标记)是长方体(用虚线框显示)。

拥挤距离计算要求在上升过程中根据每个目标函数值对人口进行排序数量级。此后，对于每个目标函数，边界解(具有最小和最大函数的解值)被赋予无限距离值。所有其他中间解都被赋予一个距离值，该距离值等于两个相邻解的函数值的绝对归一化差。这一计算与其他目标一起继续功能。总拥挤距离值计算如下与每个目标相对应的单个距离值的总和。在计算之前，每个目标函数都是标准化的拥挤的距离。算法如下所示该页概述了拥挤距离的计算过程在非支配集中的所有解中。

这里，是指集合中的个人和参数目标函数的最大值和最小值。这个过程的复杂性取决于排序算法。由于涉及最多解的独立排序(当所有种群成员都在一个前沿时)，上述算法具有计算性复杂性在集合中的所有填充成员被分配一个距离度量，我们可以比较两个解的范围与其他解决方案接近。用较小的从某种意义上说，这个距离测量值更拥挤通过其他解决方案。这正是我们在建议的拥挤比较运算符，如下所述。尽管图 1 示出了拥挤距离的计算对于两个目标，本程序适用于两个以上目标也是。

拥挤比较运算符：拥挤比较运算符(\prec)在各个算法朝着均匀分布的 `paretooptimal` 前沿发展的阶段。假设人口中的每一个人有两个属性：1) 非支配秩(\prec)；2) 拥挤距离(\prec)。我们现在定义一个偏序为。

有内容

也就是说，在具有不同非支配性的两个解之间等级，我们更喜欢等级较低(较好)的解决方案。否则，如果两个解决方案属于同一战线，那么我们更喜欢位于较不拥挤区域的解决方案。有了这三项新发明排序过程，一个快速拥挤距离估计过程，和一个简单的拥挤比较操

作，我们现在准备好描述 NSGA-II 算法。

C.主回路

最初，会创建一个随机父种群。人口按非支配性排序。每个解决方案是分配一个与其非支配等级（1）相等的适应度（或等级）是最好的级别，2 是下一个最好的级别，依此类推）。因此，假设适应度最小化。首先，使用常用的二元竞赛选择、重组和变异算子创建一个庞大的后代群体。自精英主义以来

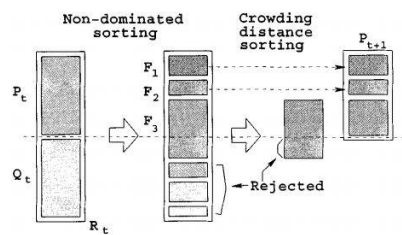


图 2。NSGA-II 程序。

中的解决方案数对于每组初始化距离为每个目标使用每个目标值排序排序以便始终选择边界点对于所有其他点。通过比较当前人口和以前的人口找到最佳的非支配解，过程不同在最初的一代之后。我们首先描述第三代如页面底部所示。逐步过程表明，nsga-ii 算法是简单明了。首先，人口总数形成。人口众多。然后，人口按非支配性排序。既然所有以前和现在的人口成员包括在，精英主义得到保证。现在，属于最佳非支配集的解是组合总体中的最佳解，必须比任何其他解更加强调总人口。如果的大小小于，我们一定会为新的人口挑选所有的成员。剩下的人口从随后的非支配战线中选出他们的排名。因此，接下来从集合中选择解，然后是集合的解，等等。这个程序一直持续到不能容纳更多的设备为止。说集合是最后一个非支配集合可容纳一套。一般来说，在从到的所有集合都将大于人口大小。为了准确地选择人口成员，我们对解决方案进行排序使用拥挤的比较运算符按降序排列并选择所需的最佳解决方案所有人口位置。NSGA-II 程序也显示在图 2。新的群体大小现在用于选择、交叉和变异，以创建新的群体大小的需要注意的是，我们使用的是二进制锦标赛选择运算符，但选择条件现在基于拥挤的比较运算符。因为这个操作员需要在种群中的每个解的秩和拥挤距离，我们在形成种群时计算这些量，如上面的算法所示。考虑整个算法一次迭代的复杂性。基本操作及其最坏情况的复杂性具体如下：

- 1) 非支配排序；
- 2) 拥挤距离分配为；
- 3) 对 IS 进行分类。

算法的总体复杂度为

由算法的非支配排序部分控制。仔细执行，整个人口规模需要不按非支配性排序。一旦分类程序已找到足够数量的前端以容纳成员，没有理由继续执行排序程序。引入了非支配解的多样性使用拥挤比较程序锦标赛选择与人口减少相位。因为解决方案与它们的拥挤距离竞争（A 领域中解的密度的度量），没有额外的刻痕参数（如 NSGA 中所需）是必需的。虽然拥挤距离是在目标函数空间中计算的，但如果需要的话，也可以在参数空间中实现。然而，在本研究中进行的所有模拟中，我们都使用了目标函数空间小生境。

```

crowding-distance-assignment( $I$ )
 $l = |I|$ 
for each  $i$ , set  $I[i]_{\text{distance}} = 0$ 
for each objective  $m$ 
     $I = \text{sort}(I, m)$ 
     $I[1]_{\text{distance}} = I[l]_{\text{distance}} = \infty$ 
    for  $i = 2$  to  $l - 1$ 
         $I[i]_{\text{distance}} = I[i]_{\text{distance}} + (I[i + 1].m - I[i - 1].m) / (f_m^{\max} - f_m^{\min})$ 

```

四、仿真结果

在本节中，我们首先描述用于比较 NSGA-II 与 PAES 和 SPEA 的性能。对于 paes 和 spea，我们有相同的参数设置正如最初的研究所建议的那样。对于 NSGA-II，我们有选择了一套合理的价值观找到最佳参数设置。我们把这个任务留给未来的研究。结合亲子种群——所有非主导的战线和直到父种群填满——计算拥挤距离在父 pop 中包含第个非支配前检查下一个前部是否包含黑变种降序使用选择的第一个元素—使用选择、交叉和变异来创建新人口增加生成计数器。

```

 $R_t = P_t \cup Q_t$ 
 $\mathcal{F} = \text{fast-non-dominated-sort}(R_t)$ 
 $P_{t+1} = \emptyset$  and  $i = 1$ 
until  $|P_{t+1}| + |\mathcal{F}_i| \leq N$ 
    crowding-distance-assignment( $\mathcal{F}_i$ )
     $P_{t+1} = P_{t+1} \cup \mathcal{F}_i$ 
     $i = i + 1$ 
sort( $\mathcal{F}_i, \prec_n$ )
 $P_{t+1} = P_{t+1} \cup \mathcal{F}_i[1 : (N - |P_{t+1}|)]$ 
 $Q_{t+1} = \text{make-new-pop}(P_{t+1})$ 

 $t = t + 1$ 

```

deb 等人：一个快速的精英多目标遗传算法：nsga-ii

TABLE 1 List of the 10 models of the form (1.1)					
Problem	Variables	Equations	Boundary conditions	Comments	
PROB 1	u, v, w	(1.1a)–(1.1c)	$u = v = w = 0$ on $\partial\Omega$	classical	
PROB 2	u, v	(1.1a)–(1.1b)	$u = v = 0$ on $\partial\Omega$	classical	
PROB 3	u, v, w	(1.1a)–(1.1c)	$u = v = 0$ on $\partial\Omega$	classical	
PROB 4	u, v, w	(1.1a)–(1.1c)	$u = v = 0$ on $\partial\Omega$	classical	
PROB 5	u, v, w	(1.1a)–(1.1c)	$u = v = 0$ on $\partial\Omega$	classical	
PROB 6	u, v, w	(1.1a)–(1.1c)	$u = v = 0$ on $\partial\Omega$	classical	
PROB 7	u, v, w	(1.1a)–(1.1c)	$u = v = 0$ on $\partial\Omega$	classical	
PROB 8	u, v, w	(1.1a)–(1.1c)	$u = v = 0$ on $\partial\Omega$	classical	
PROB 9	u, v, w	(1.1a)–(1.1c)	$u = v = 0$ on $\partial\Omega$	classical	
PROB 10	u, v, w	(1.1a)–(1.1c)	$u = v = 0$ on $\partial\Omega$	classical	

A、测试问题

我们首先描述用于比较 MOEAs。测试问题是从过去在这个领域的一些重要研究中挑选出来的。Veldhuizen[22]引用了一个数字过去使用过的测试问题。他们，我们选择四个问题：schaffer 的研究 (sch) [19]，fonseca 和弗莱明的研究 (fon) [10]，波罗尼的研究 (pol) [16]，以及库尔索的研究 (库尔) [15]。1999 年，第一作者建议多目标优化测试问题的系统化方法[3]。Zitzler 等人[25]遵循这些指导原则，提出了六个测试问题。我们选了五个这里有六个问题，叫做 zdt1, zdt2, zdt3, zdt4, 和 ZDT6。所有问题都有两个目标函数。没有这些问题中有任何限制。我们在表一中描述了这些问题。表中还显示了变量的数量，它们的边界，帕累托最优解，以及每个问题的帕累托最优前沿。

所有进近都以 25000 的最大功能运行评价。我们使用单点交叉和按位二进制编码气体的变异与模拟二进制交叉 (sbx) 算子与多项式变异[6]实数编码气体。和的交叉概率或 (其中实数编码气体的决策变量数使用二进制编码气体的字符串长度)。对于实数编码 NSGA-II，我们使用分布指数[6]进行交叉和变异算子分别为和。在 250 代人之后获得的人口应用精英保留算子后的种群) 用于计算两个性能指标，我们将讨论在下一节。对于 paes，我们使用一个等于四个，档案馆大小 100。我们利用所有人口在 25000 次迭代结束时获得的存档成员计算性能指标。对于 spea，我们使用人口 80 和外部人口 20 (这个 4:1 比例由 SPEA 开发商建议维持为精英解决方案提供足够的选择压力)，以便总人口规模变为 100 人。spea 也运行到完成 25000 项功能评估。对于 spea，我们使用遗传算法与外部算法结合的非支配解计算性能的最后世代种群本研究中使用的指标。对于 paes、spea 和二进制编码 NSGA-II，我们使用了 30 位来编码每个决策变量。

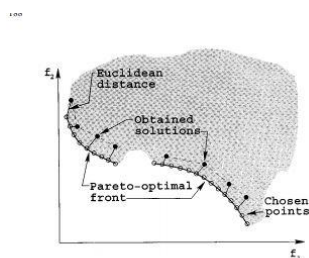


Fig. 3. Distance metric Y .

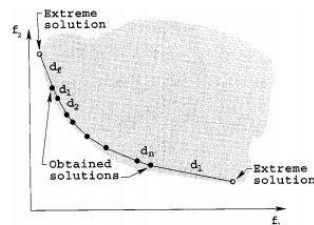


Fig. 4. Diversity metric Δ .

B. 业绩衡量

与单目标优化不同，在一种多目标优化方法：1) 收敛于 pareto 最优集，2) 保持解的多样性。帕累托最优集。这两项任务不能用一个性能指标来充分衡量。已经提出了许多性能指标[1]、[8]、[24]。在这里，我们定义了两个性能指标，它们在评估多目标解集上的两个

目标优化算法。

第一个度量收敛到已知 pareto 最优解集。因为多目标算法将在一组已知的帕累托最优解，这个度量的计算是可能的。然而，我们意识到，这样的标准不能被使用任何武断的问题。首先，我们从真 pareto 最优前沿发现了一组均匀分布的解。目标空间。对于用一个算法得到的每个解，我们计算它的最小欧氏距离帕累托最优前沿上的选择解。平均值其中的距离用作第一个度量（收敛度量）。图 3 显示了这种方法的计算过程。阴影区域是可行的搜索区域，而实心曲线表示 pareto 最优解。在 pareto 最优前沿上选择开圆解来计算收敛性度量，而带黑圈的解是由算法。这个度量值越小，越好向帕累托最优前沿收敛。

当全部获得解决方案完全取决于所选的解决方案，此度量采用值为零。在这里执行的所有模拟中，我们将展示为求解而计算的此度量的平均值和方差多次运行获得的集合。即使所有解都收敛到帕累托最优前沿，上述收敛度量的值不为零。这个只有当每一个获得的解恰好位于每一个选择的解上时，度量才会产生零。尽管仅此指标。我们定义了一个不同的度量来度量直接由算法得到的解。第二度量测量获得的解决。在这里，我们有兴趣得到一套解决方案它跨越了整个帕累托最优区域。我们计算得到的非支配解集中连续解之间的欧氏距离。我们计算平均值这些距离。然后，从得到的非支配解集中，我们首先计算出（目标空间）通过拟合一条与真实空间平行的曲线帕累托最优前沿。然后，我们使用以下度量来计算分布中的不均匀性：

$$\Delta = \frac{d_f + d_t + \sum_{i=1}^{N-1} |d_i - \bar{d}|}{d_f + d_t + (N-1)\bar{d}}.$$

这里，参数和是极限解和边界解之间的欧几里德距离得到非支配集，如图 4 所示。该图说明了上述方程式中提到的所有距离。参数是所有距离的平均值，假设在最佳非支配前沿有解决方案。有了解决方案，距离。分母是当所有解都在一个解上时。有趣的是注意，这并不是最坏情况下可能的解决方案。我们可以有一个场景，其中有一个很大的差异。在这种情况下，度量值可能大于 1。因此，上述度量的最大值可以大于 1。然而，良好的分布会使所有距离相等使非支配集中的解。因此，对于最广泛的均匀分布一组非支配解，其分子为零，使得度量取一个值。零。对于任何其他分布，度量值将是大于零。对于具有相同值的两个分布和的度量取一个更高的值，并且在极端解内解的分布更差。请注意上述分集度量可用于解，包括不是帕累托最优集的解。

Algorithm	SCH	FON	POL	KUR	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
NSGA-II	0.003391	0.001931	0.015553	0.028964	0.033482	0.072391	0.114500	0.513053	0.296564
Real-coded	0	0	0.000001	0.000018	0.004750	0.031689	0.007940	0.118400	0.013135
NSGA-II	0.002833	0.002571	0.017029	0.028951	0.000894	0.000824	0.043411	3.227636	7.806798
Binary-coded	0.000001	0	0.000003	0.000016	0	0	0.000342	7.307653	0.001667
SPEA	0.003403	0.125692	0.037812	0.045617	0.001799	0.001339	0.047517	7.340299	0.221138
	0	0.000038	0.000088	0.000005	0.000001	0	0.000047	6.572516	0.000449
PAES	0.001313	0.151263	0.030864	0.057323	0.082095	0.126276	0.023872	0.854816	0.085469
	0.000003	0.000905	0.000431	0.011989	0.008979	0.036877	0.00001	0.527238	0.006664

Algorithm	SCH	FON	POL	KUR	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
NSGA-II	0.477899	0.378065	0.452150	0.411477	0.390387	0.430716	0.730540	0.702612	0.668025
Real-coded	0.003471	0.000639	0.002860	0.000992	0.001376	0.004721	0.019706	0.064648	0.009923
NSGA-II	0.449265	0.395131	0.503721	0.442195	0.463292	0.435112	0.575606	0.479475	0.644477
Binary-coded	0.002052	0.001314	0.004656	0.001498	0.041622	0.024607	0.005078	0.009841	0.035042
SPEA	1.021110	0.792352	0.972783	0.852990	0.784525	0.755148	0.672938	0.798463	0.849389
	0.004372	0.005546	0.008475	0.002619	0.004440	0.004521	0.003587	0.014616	0.002713
PAES	1.063288	1.162528	1.020007	1.079838	1.229794	1.165942	0.769920	0.870458	1.153052
	0.002688	0.008945	0	0.013772	0.004839	0.007682	0.001653	0.101399	0.003916

使用三角化技术或 voronoi 图方法[1]为了计算，可以将上述过程扩展到估计在更高维度中解的扩散。

C. 讨论结果

表二显示了收敛的均值和方差使用四种算法 NSGA-II（实数编码）获得的度量，NSGA-II（二进制编码）、SPEA 和 PAES。nsga-ii（实数编码或二进制编码）能够收敛除了 ZDT3 和 ZDT6，所有问题都比 PAES 好找到了更好的融合点。在所有使用 NSGA-II 的情况下，10 次运行的方差也很小，但使用 NSGA-II 的 ZDT4 除外（二进制编码）。paes 的固定归档策略允许更好的九个问题中有两个需要收敛。表三显示了多样性度量的均值和方差使用所有三种算法获得。NSGA-II（实数或二进制编码）在所有九种编码中表现最好测试问题。paes 表现最差。为了举例说明，我们在图 5 中的问题 sch 上显示了十次 paes 运行中的一次，其中一次运行了任意一次 nsga-ii（实数编码）。在大多数问题上，实数编码的 nsga-ii 能够找到比任何其他算法都能更好地传播解，包括二进制编码的 NSGA-II。为了演示这些算法的工作原理，我们还展示了 paes、spea 和 NSGA-II 关于测试问题 kur、zdt2、zdt4 和 zdt6。问题 kur 在帕累托最优前沿。图 6 显示了所有非支配解使用 NSGA-II（实数编码）在 250 代后获得。这个图中还显示了帕累托最优区域。这个数字展示 NSGA-II 向真实世界收敛的能力在前面寻找不同的解决方案。图 7 显示得到了具有 spea 的非支配解，即这个问题的次优算法（参见表二和表三）。

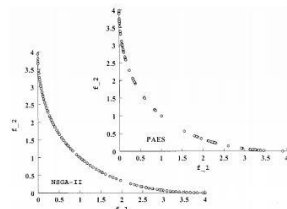


Fig. 5. NSGA-II finds better spread of solutions than PAES on SCH.

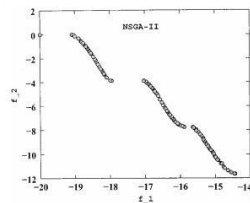


Fig. 6. Non-dominated solutions with NSGA-II (real-coded) on KUR.

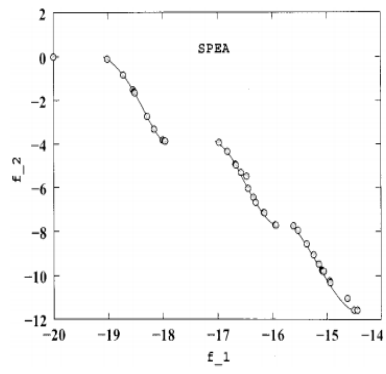


Fig. 7. Nondominated solutions with SPEA on KUR.

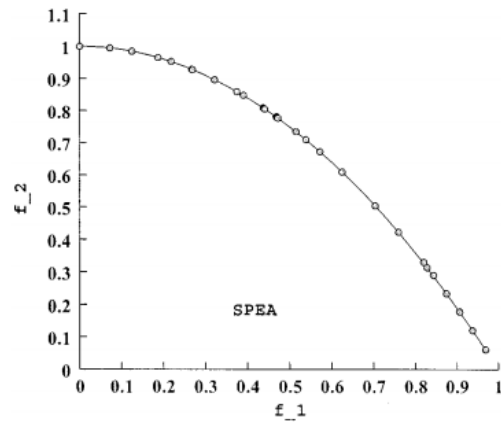


Fig. 9. Nondominated solutions with SPEA on ZDT2.

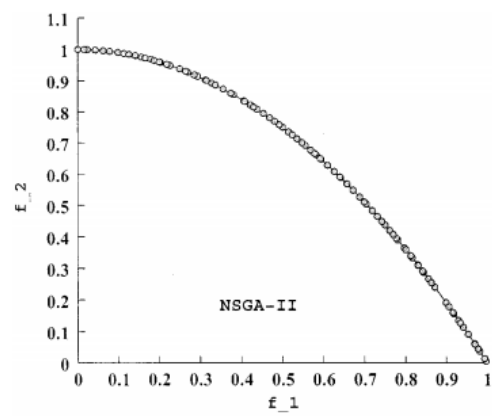


Fig. 8. Nondominated solutions with NSGA-II (binary-coded) on ZDT2.

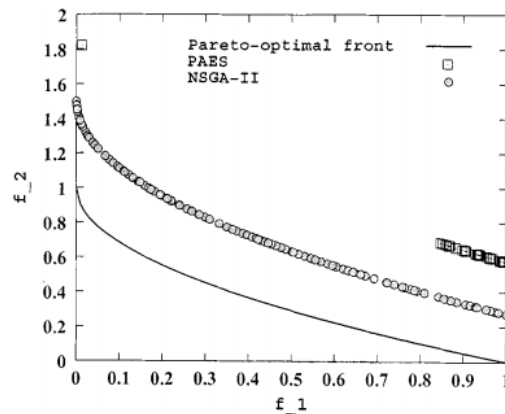


Fig. 10. NSGA-II finds better convergence and spread of solutions than PAES on ZDT4.

在解决办法的趋同和分配两方面，nsga-ii 在这个问题上比 spea 表现得更好。

自 spea 无法在最终的 ga 族，非支配的总数与最终结果相比，解决方案要少得多 nsga-ii 人口。

接下来，我们将展示关于这个问题的非主导性解决方案

图 8 和图 9 中的 zdt2。这个问题有一个非凸的对开时间线前。我们展示二进制编码 nsga-ii 的性能以及这个功能。尽管收敛不是两个算法都有困难，包括真实的-和二进制编码的 nsga-ii 已经发现了一个更好的传播和更多在整个帕累托最优区域的解比 spea（针对这个问题观察到的 Next-Best 算法）。

zdt4 的问题有 21 或 7.94(10) 不同的局部搜索空间中最理想的前端，其中只有一个对应于全球帕累托最优战线。· 欧几里德人两个安全局部帕累托最优集的解之间的决策空间距离为 0.25。图五。10 表明实际编码的 nsga-ii 和 paes 都陷入了不同的困境局部帕累托最优集合，但收敛性和能力找到一套不同的解决方案肯定是更好的 nsga-II。二进制编码气体难以收敛在全球帕累托-最佳战线附近，也是一个问题在以前的单目标研究中观察到的[5]。在类似的

Rastrigin 的函数[这里的函数]，这项研究清楚地表明，人口规模约为单目标二值编码气体至少需要 500（有赛事选择，单点交叉和位元 mutation），在超过 50% 的模拟运行。· 因为我们使用了在 100 码的情况下，不可能发现全球帕累托最优解决方案，但是 nsga-ii 能够找到即使是在当地的帕累托最优阵线，解决方案的良好传播。由于 Spea 在这一问题上意见不一致（见表二），我们没有显示这个数字的详细结果。最后，图 11 显示，spea 发现一个更好的聚合与其他任何解决方案相比，zdt6 中的一组非主导解决方案算法。

然而，在解决方案中的分布更好的是真正编码的 `nsga-ii`。

d. 不同的参数设置

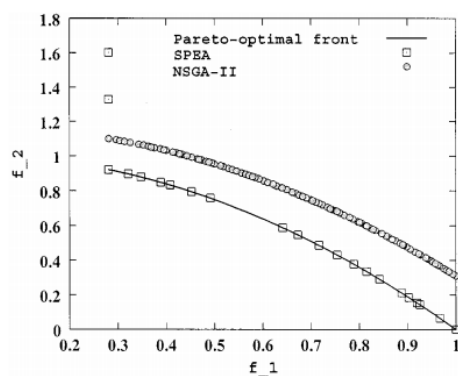


Fig. 11. Real-coded NSGA-II finds better spread of solutions than SPEA on ZDT6, but SPEA has a better convergence.

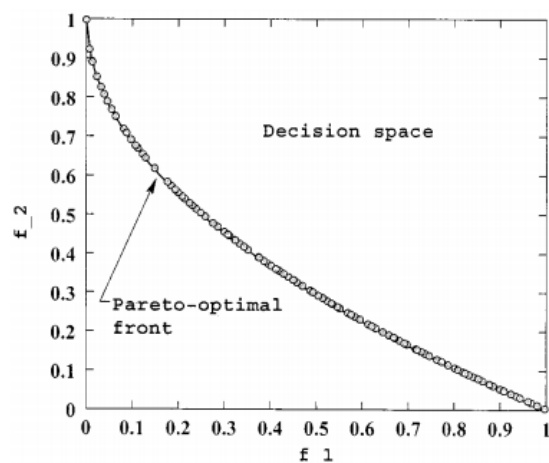


Fig. 12. Obtained nondominated solutions with NSGA-II on problem ZDT4.

Convergence metric, Υ					
	POL	KUR	ZDT3	ZDT4	ZDT6
Mean	0.015882	0.026544	0.018510	0.090692	0.276609
Variance	0.000001	0.000017	0.000227	0.053460	0.015843
Diversity metric, Δ					
	POL	KUR	ZDT3	ZDT4	ZDT6
Mean	0.467022	0.418889	0.688218	0.440022	0.655896
Variance	0.002186	0.000530	0.000610	0.026729	0.003302

在本研究中，我们并没有认真尝试找出 `nsga-ii` 的最佳参数设置。但在这一节，我们做了额外的实验来证明 `nsga-ii` 性能的不同参数设置。首先，我们保留所有其他参数，但增加 500 代（而不是所用的 250 代）以前）。表四显示了收敛性和多样性指标针对问题 `Pol`, `Kur`, `zdt3`, `zdt4` 和 `zdt6`。

现在，我们实现非常接近真正的帕累托最优锋的收敛有更好的分配。表中显示了这些难题，真正编码的 `nsga-ii` 已经收敛了很接近真正的最佳战线，除了 `zdt6`，现在，我们实现非常接近真正的帕累托最优锋的收敛有更好的分配。表中显示了这些难题，真正编码的 `nsga-ii` 已经收敛了很接近真正的最佳阵线，除了 `zdt6`，它可以要求用 `nsga-ii` 设置不同的参数。通常情况下，`zdt3` 和 `zdt4` 的结果随属数的增加而改善。

`zdt4` 问题有一些局部帕累托最优前端，每个对应于特定值的。一大块需要改变决策向量才能离开本地擎天柱除非突变或交叉运算符有能力在另一个更好的吸引者的盆地里创造解决方案，对真正的帕累托时间线的收敛性的改进是不可能的。我们使用 `nsga-ii` (实编码) 变异的较小分布指数创建比以前更广泛的解决方案的效果。休息参数设置与以前相同。`zdt4` 问题有一些局部帕累托最优前端，每个对应于特定值的。一大块需要改变决策向量才能离开本地擎天柱除非突变或交叉运算符有能力在另一个更好的吸引者的盆地里创造解决方案，对真正的帕累托时间线的收敛性的改进是不可能的。我们使用 `nsga-ii` (实编码) 变异的较小分布指数创建比以前更广泛的解决方案的效果休息参数设置与以前相同。关于问题 `zdt4` 的康维尔度量和多样性度量。

250 代的结束如下：

$$\begin{array}{ll} \overline{\Upsilon} = 0.029544 & \sigma_{\Upsilon}^2 = 0.002145 \\ \overline{\Delta} = 0.498409 & \sigma_{\Delta}^2 = 0.003852. \end{array}$$

如图所示，这些结果比 `paes` 和 `spea` 好得多表二。为了证明这种音乐的收敛性和传播性，我们绘制了其中一种音乐的非控制解在图 12 的 250 代之后。图为 `nsga-ii` 能够在真正的帕雷托-最佳战线上找到解决方案。

五、轮换问题

以前的一项研究[3]讨论过相互作用在决策变量中，任何多目标优化算法都可以引入另一个层次的不确定性，包括艾斯在本节中，我们提出了一个这样的问题，并研究了前三个步骤在以下方面的工作认识论问题：

$$\begin{aligned}
 &\text{minimize} && f_1(\mathbf{y}) = y_1 \\
 &\text{minimize} && f_2(\mathbf{y}) = g(\mathbf{y}) \exp(-y_1/g(\mathbf{y})) \\
 &\text{where} && g(\mathbf{y}) = 1 + 10(n-1) \\
 &&& + \sum_{i=2}^n [y_i^2 - 10 \cos(4\pi y_i)] \\
 &\text{and} && \mathbf{y} = \mathcal{R}\mathbf{x} \\
 &&& -0.3 \leq x_i \leq 0.3, \quad \text{for } i = 1, 2, \dots, n.
 \end{aligned} \tag{2}$$

ea 与决策变量向量工作，但上述客观函数是用变量向量定义的，它是通过转换决策变量向量来计算的由一个固定的旋转矩阵。这样，目标函数是决策变量线性组合的函数。元素钢的符号为了在帕累托最优化上保持解决方案的分布区域甚至收敛到任何特定的解决方案都需要以特定方式更新所有决策变量。元素钢的符号为了在帕累托最优化上保持解决方案的分布区域甚至收敛到任何特定的解决方案都需要以特定方式更新所有决策变量。你可以用这个词。一种通用的搜索运算符，如变量 sbx 运算符在这里，这对艺电来说是一项艰巨的任务。然而，在这里，我们有兴趣评估三个精英莫厄斯。

我们使用 100 的人口规模并运行每个算法直到 500 代。对于 sbx，我们使用和使用为了突变。将帕累托最优解决方案限制在在规定的变量范围内，我们不鼓励解决方案通过增加一个固定的大处罚两个 objec。

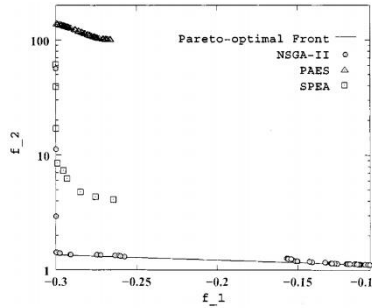


Fig. 13. Obtained nondominated solutions with NSGA-II, PAES, and SPEA on the rotated problem.

13 显示 500 年底得到的解使用 nsga-ii、paes 和 spea 的世代。向帕累托最优方向前进所需的相关的 PA 刻度表更新前线使这种问题难以解决，在 500 结束时显示得到的解使用 nsga-ii、paes 和 spea 的世代。它被观察到 nsga-ii 解决方案比较接近真正的领域由 paes 和 spea 得到的解。向帕累托最优方向前进所需的相关参数更新前线使这种问题难以解决。nsga-ii。

与实编码交叉操作一起保存精英的操作符而突变算子能够找到一些接近[由此产生的]帕累托-最优阵线。这个示例问题表明，单目标优化算法的已知困难之一（链接问题[11]、[12]）也会在多目标问题中造成困难然而，需要进行更系统的研究，以便解决多目标优化中的关联问题。

六、约束处理

在过去，第一作者和他的学生实施了单目标优化的无约束处理方法。那些研究[2]，[6]表明了基于锦标赛选择的算法可以用来处理人口方法中的制约因素要比数字好得多其他现有的约束处理方法。可以用上述 nsga-ii 采用类似的方法来解决约束的多目标优化问题。

a. 拟议的约束处理办法 nsga-ii

这种约束处理方法使用二进制比赛选择，从人群中选取两个解决方案选择更好的解决方案。在有限制的情况下，每个解决方案可以是可行的，也可以是不可行的。因此，这里可能最多有三种情况：1)两种解决办法都是可行的；一是可行，二是行不通；三是两者都不可行。对于单目标优化，我们使用了一个简单的规则事例。

案例 1：选择目标函数更好的解价值。

案例 2：选择可行的解决方案。

案例 3：选择整体约束较小的解决方案违规行为。

因为在任何情况下，约束和目标函数值都不是与彼此相比，没有必要有任何惩罚参数，使所建议的约束处理方法有用和有吸引力。

在多目标优化的背景下，后两者案件可以按原样使用，第一个案件可以通过像以前一样使用人群比较运算符。以维持 nsga-ii 过程中的模块化，我们只是简单地修改两个解决方案之间的支配性的定义。

定义 1：一个解决方案被称为约束支配解，如果下列任何一个条件为真。

- 1)解决是可行的，解决不是。
- 2)解决办法都是不可行的，但解决办法有较小的整体约束违约。
- 3)解决方案是可行的，解决方案占主导地位解决方案。

使用这种约束支配原则的效果任何可行的解决方案都有更好的非支配等级比任何不可行的解决办法都重要。所有可行的解决方案都被排序根据他们的非统治水平基于目标函数值。然而，在两个不可行的解决方案中。

使用较小的约束违反的解决方案具有更好的级别。此外，这种对非支配原则的修改不改变 nsga-ii 的计算复杂度。如前所述，nsga-ii 程序的其余部分可以是像往常一样使用。

他以上的约束统治的定义与用 fonseca 和 freming 建议[9]。唯一的区别就是以这种方式为不可行的解决方案定义了支配。在上面的定义中，一个不可行的解决方案，具有较大的总体限制---违反行为被归类为更大的非支配等级。另一方面，在[9]中，违反不同限制的不可行解决方案被归类为成员同样不受控制的前线。因此，一个不可行的解决方案轻微违反约束将被置于同一另一个解决方案违反了在很大程度上是制约因素。这可能导致算法在通过约束边界到达可行区域之前，在不可行的搜索区域中漫游更多代。此外，由于 fonseca--弗莱明的方法要求以约束违反值进行控制检查，建议本文的方法在计算上较不昂贵，而且更简单。

b .Seow 的约束处理方法

ray 等人[17]提出了一种更详细的约束处理技术，在这种技术中，约束违反了所有约束不是简单地归纳在一起。相反，非统治还检查了违反限制的情况。我们给出一个大纲在这里做这个手术。

表五本研究所用的约束测试问题

TABLE V
CONSTRAINED TEST PROBLEMS USED IN THIS STUDY

Problem	n	Variable bounds	Objective functions	Constraints
CONSTR	2	$x_1 \in [0.1, 1.0]$ $x_2 \in [0, 5]$	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = (1 + x_2)/x_1$	$g_1(\mathbf{x}) = x_2 + 9x_1 \geq 6$ $g_2(\mathbf{x}) = -x_2 + 9x_1 \geq 1$
SRN	2	$x_i \in [-20, 20]$ $i = 1, 2$	$f_1(\mathbf{x}) = (x_1 - 2)^2 + (x_2 - 1)^2 + 2$ $f_2(\mathbf{x}) = 9x_1 - (x_2 - 1)^2$	$g_1(\mathbf{x}) = x_1^2 + x_2^2 \leq 225$ $g_2(\mathbf{x}) = x_1 - 3x_2 \leq -10$
TNK	2	$x_i \in [0, \pi]$ $i = 1, 2$	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = x_2$	$g_1(\mathbf{x}) = -x_1^2 - x_2^2 + 1 + 0.1 \cos(16 \arctan(x_1/x_2)) \leq 0$ $g_2(\mathbf{x}) = (x_1 - 0.5)^2 + (x_2 - 0.5)^2 \leq 0.5$
WATER	3	$0.01 \leq x_1 \leq 0.45$ $0.01 \leq x_2 \leq 0.10$ $0.01 \leq x_3 \leq 0.10$	$f_1(\mathbf{x}) = 106780.37(x_2 + x_3) + 61704.67$ $f_2(\mathbf{x}) = 3000x_1$ $f_3(\mathbf{x}) = (305700)2289x_2/(0.06 \times 2289)^{0.65}$ $f_4(\mathbf{x}) = (250)2289 \exp(-39.75x_2 + 9.9x_3 + 2.74)$ $f_5(\mathbf{x}) = 25(1.39/(x_1x_2) + 4940x_3 - 80)$	$g_1(\mathbf{x}) = 0.00139/(x_1x_2) + 4.94x_3 - 0.08 \leq 1$ $g_2(\mathbf{x}) = 0.000306/(x_1x_2) + 1.082x_3 - 0.0986 \leq 1$ $g_3(\mathbf{x}) = 12.307/(x_1x_2) + 49408.24x_3 + 4051.02 \leq 50000$ $g_4(\mathbf{x}) = 2.098/(x_1x_2) + 8046.33x_3 - 696.71 \leq 16000$ $g_5(\mathbf{x}) = 2.138/(x_1x_2) - 7883.39x_3 - 705.04 \leq 10000$ $g_6(\mathbf{x}) = 0.417(x_1x_2) + 1721.26x_3 - 136.54 \leq 2000$ $g_7(\mathbf{x}) = 0.164/(x_1x_2) + 631.13x_3 - 54.48 \leq 550$

All objective functions are to be minimized.

三种不同的非主导人口排名为先演出。第一个排序使用目标函数值执行，并将生成的排序存储在一个维向量中。进行第二次排名只使用所有约束的约束违约值，不使用目标函数信息。因此，约束违反每个约束被使用为一个标准，并且对人口进行非支配性分类带有约束违约值。注意，对于一个可行的解决所有的约束违反是零。因此，所有可行的解决方案都有一级。完成第三顺序论客观功能与约束侵权行为的结合价值[价值总数]。这就产生了排名。虽然目标函数值和约束违逆一起使用，但该算法的一个很好的方面是不需要任何惩罚参数一旦这些排名结束，所有排名最好的可行解决方案都会被选中为新的人群。如果有更多的人口空缺，它们是从剩余的解决方案系统地创建的。通过重视在选拔经营者中的排名，重视在选拔经营者中的排名。交叉算子，研究者们提出了一个系统的多目标 ga，其中也包括一个维护精细算子。有关详情，读者可参阅[17]。尽管调查人员没有将他们的算法与其他方法进行比较，他们展示了这种约束处理方法的工作原理。

一些工程设计上的问题。但是，自从需要对三组不同的准则进行非主导排序，该算法引入了许多不同的运算符，它在更复杂的环境中的表现有待研究问题，特别是从计算的角度与方法相关的负担。在下一节中，我们选择一组四个问题，并将简单约束的 nsga-ii 与 Seow 的方法。

C. 模拟结果

我们选择四个约束测试问题（见表五）在早期的研究中使用过。在第一个问题中，无约束帕累托优化区域是不可行的。因此，所得到的约束帕累托最优区域是第一个约束边界和无约束帕累托最优区域的一部分的串联。第二个问题是用于 nsga[20]的原始研究。在这里，被约束的人帕累托最优集是无约束帕累托最优集的子集。第三个问题是田中等人提出的。Al. [21]并且有一个不连续的帕累托最优区域，正在下降完全依赖于第一个约束边界。在下一节中，我们给出了每个以上问题。第四个问题水是一个五重七重的问题，试图在[17]中解决。有了五个目标，就很难讨论无约束帕累托最优区域的约束。联合国系统下一节，我们将展示所有或十幅获得的配对图非主导的解决方案。我们在这里应用了真实编码的 nsga-ii。

在所有的问题中，我们使用 100 的种群大小，分布指数为实编码交叉和突变算子分别为 20 和 100，并运行 nsga-ii（真实编码）与所提出的约束处理技术和。我们选这个比较大有多少代要调查的是解决方案中的传播是否能维持很多代。然而，在每种情况下，我们都得到了相当好的解决方案早于 200 代。Seow 的约束处理算法[17]最多 500 代。交叉和突变概率和以前一样。

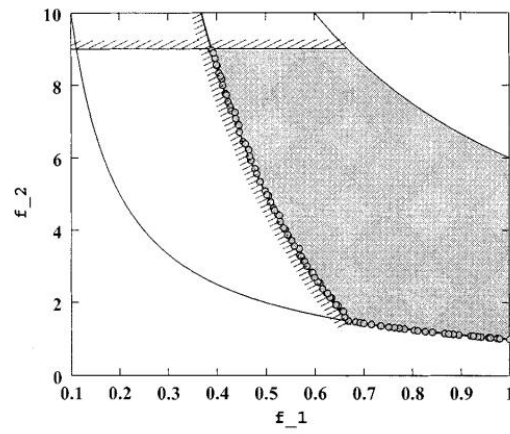


Fig. 14. Obtained nondominated solutions with NSGA-II on the constrained problem CONSTR.

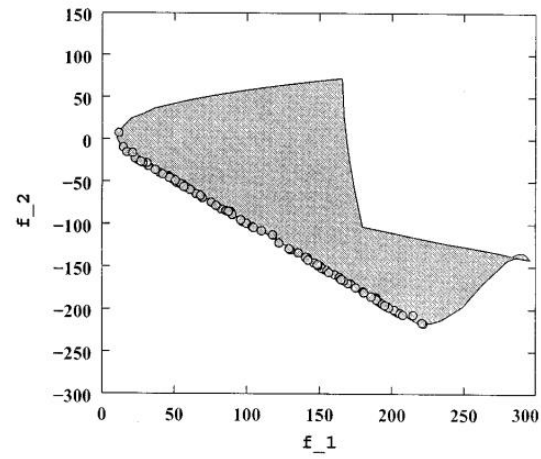


Fig. 17. Obtained nondominated solutions with Ray-Tai-Seow's algorithm on the constrained problem SRN.

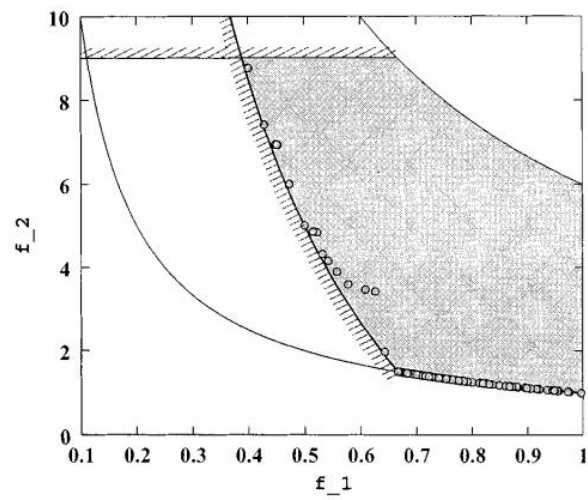


Fig. 15. Obtained nondominated solutions with Ray-Tai-Seow's algorithm on the constrained problem CONSTR.

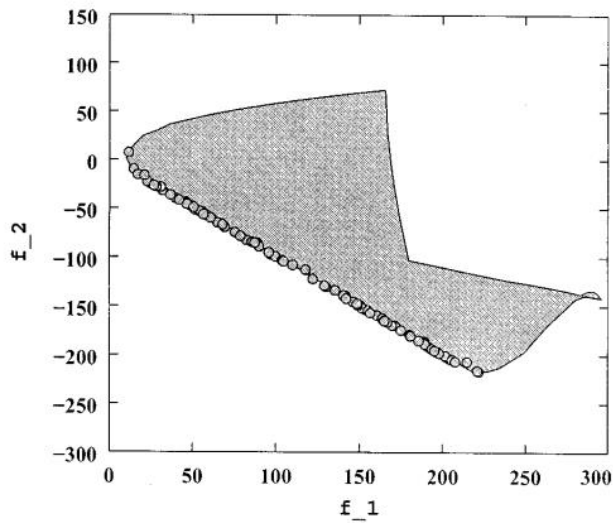


Fig. 17. Obtained nondominated solutions with Ray-Tai-Seow's algorithm on the constrained problem SRN.

图 14 显示了在 500 代使用 `nsga-ii` 之后获得的 100 套非主导解决方案。如图所示 `nsga-ii` 能够在两种方法中统一维持解决方案最理想的区域。必须指出的是, 为了实现这一目标, 在约束边界上保持解决方案的分布, 必须以约束函数规定的特定方式修改解决方案。这成为一项艰巨的任务。任何搜索人员。图 15 显示了使用 Ray-Tai-Seow 的算法在 500 代之后。很明显 `nsga-ii` 比 Seow 的算法在向真正的帕累托最优锋收敛的条件以及在维持多样化的非主导人口方面解决方案。

接下来, 我们考虑测试问题。图 16 显示了 500 代后使用 `nsga-ii` 的非主导解决方案。图为 `nsga-ii` 如何带来随机种群在帕累托-最佳阵线。Seow 的算法也是能够接近前面的这个测试问题(图 17)。图 18 和 19 显示了可行的客观空间和用 `nsga-ii` 和 Seow 的算法在这里, 帕累托最优区域是不连续的并且 `nsga-ii` 在真正的帕累托优化中找到广泛的解决方案区域。尽管 Seow 的算法找到了一个数字在帕累托-最优方面的解决方案, 有很多即使在 500 代之后也不可行的解决方案。以便演示了 `fonseca` 的约束处理策略的工作, 我们用 `nsga-ii` 实现了该策略并应用到了好吧。图五。20 显示 100 个人口成员 500 代, 使用相同的参数设置图 18。这两个数字都表明, `fonseca`—弗莱明的约束处理策略效果良好与 `nsga-ii`。

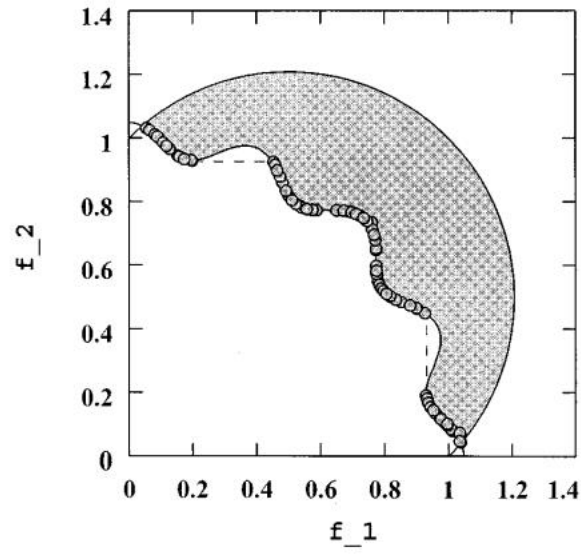


Fig. 18. Obtained nondominated solutions with NSGA-II on the constrained problem TNK.

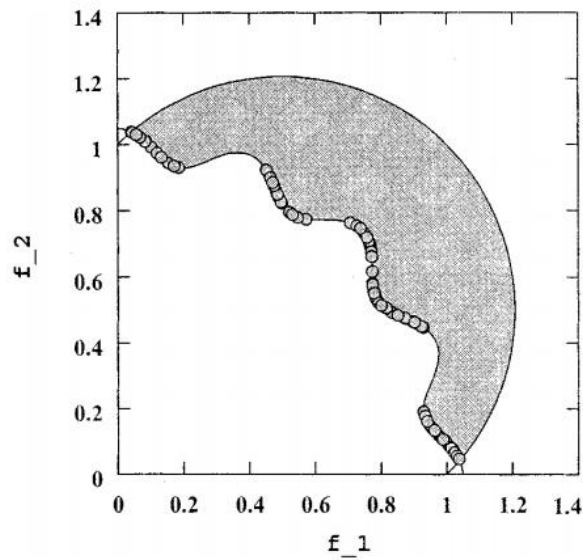


Fig. 20. Obtained nondominated solutions with Fonseca-Fleming's constraint-handling strategy with NSGA-II on the constrained problem TNK.

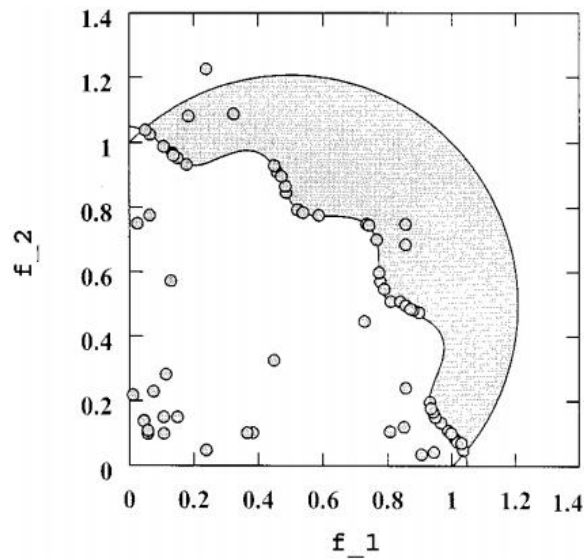


Fig. 19. Obtained nondominated solutions with Ray-Tai-Seow's algorithm on the constrained problem TNK.

雷等人[17]在他们的研究中使用了问题水。学习。它们将目标函数归一化如下曼纳：既然问题水有五个客观的功能，我们观察归一化目标函数的范围得到的非支配解的值。表六显示与 Seow 算法的比较。在大多数情况下，客观功能，nsga-ii 发现解决办法比 Seow 的方法。以显示这五个归一目标之间的对对相互作用函数，我们在图 21 中为两者绘制全部或 10 个交互图算法。nsga-ii 的结果显示在上对角线上图的部分和 Seow 的结果显示在较低的对角线部分。任何情节的轴心都可以是通过查看相应的对角线框和它们的射程。例如，第一排和第三排的图纵轴为纵轴，横轴为纵轴。由于这个图属于对角线的上部，所以这个图是使用 nsga-ii 获得的。为了比较这个情节利用雷-Seow 的方法，我们可以看到类似的情节为第三行和第一列的情节。对于这个数字，垂直轴绘制成水平轴绘制成就像。为了更好地比较这两个情节，我们观察 Seow 的情节，但把第 90 页翻到 nsga-ii 的顺时针方向结果。这将使在这两种情况下，轴的标记和范围相同。

我们观察到 nsga-ii 地块有更好的形成模式比 Seow 的阴谋还多。例如，数字-，和-相互作用是非常清楚的从 nsga-ii 结果。尽管在获得的结果中存在类似的模式使用 Seow 的算法，收敛到真边界不够。

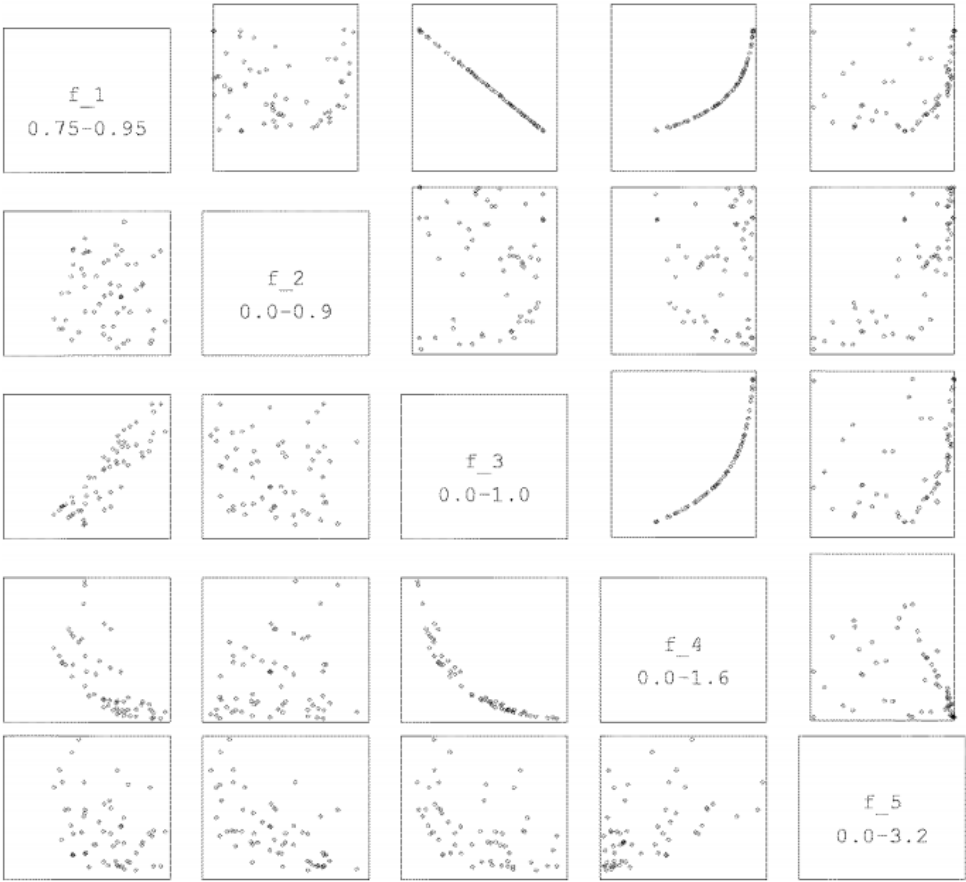
七、结论

我们提出了一种快速、精益求精的方法基于非支配排序方法。在九个不同的从文献中借鉴困难的测试问题，提出的 nsga-ii 能够更好地保持解决案的传播，并在得到的非主导前沿更好地收敛与其他两位精英莫厄斯相比一派斯和斯帕。然而，有一个问题，派斯，能够收敛得更接近真实最理想的正面。各解决方案之间保持多样性通过控制在搜索空间中相同大小的

细胞的决定性和预先指定数量的溶液拥挤。元素钢的符号在这个问题上，人们怀疑如此决定性的拥挤再加上基于变异的方法的效果与 `nsga-ii` 中使用的动态和无参数拥挤方法相比，在靠近真实战线处收敛是有益的和斯帕。然而，所使用的多样性保护机制在 `nsga-ii` 被发现是三种方法中最好的在这里学习。

TABLE VI
LOWER AND UPPER BOUNDS OF THE OBJECTIVE FUNCTION VALUES OBSERVED IN THE OBTAINED NONDOMINATED SOLUTIONS

Algorithm	f_1	f_2	f_3	f_4	f_5
NSGA-II	0.798 – 0.920	0.027 – 0.900	0.095 – 0.951	0.031 – 1.110	0.001 – 3.124
Ray-Tai-Seow	0.810 – 0.956	0.046 – 0.834	0.967 – 0.934	0.036 – 1.561	0.211 – 3.116



在参数相互作用强的问题上，`nsga-ii` 比其他人更接近真正的前线两种法，但重要的是这三种方法在解决这一所谓的高度认识论问题. 虽然这是一个正在进行的研究的问题，在单一目标评价研究，本文显示这种高度认识论的问题也可能导致莫厄斯更重要的是，这个领域的研究人员应该考虑这样的认识论问题来测试一个新开发的多目标优化算法。

我们还建议对定义进行简单的扩展约束多目标优化的优势。虽然这个新定义可以用于任

何其他的百花目，带有此定义的实编码 nsga-ii 已经显示解决四个不同的问题要比最近提出的另一种约束处理方法好得多。

具有快速非支配排序过程的属性，精英主义的策略，没有参数的方法和简单的有效的约束处理方法 nsga-ii 在不久的将来应得到越来越多的关注和应用。

参考文献

- [1]K. Deb, 使用进化算法进行多目标优化。英国奇切斯特：威利，2001 年。
- [2]，“遗传算法的有效约束处理方法，”这是我的荣幸方法。墨克[原件：英文]第 186、2、4、311 至 338 页 2000 年。
- [3]，“多目标遗传算法：问题的困难和测试函数的构建，”在 evol.《计算》，1999 年，第二卷。7,第 205-230 页。
- [4]K. Deb 和 D. E. Goldberg, “基因功能优化中的生态位和物种形成的调查”，在 J. D. Schaffer 的第三次遗传算法国际会议记录中。先生 Morgan Kauffman, 1989 年，第 42 至 50 页。
- [5]K. Deb 和 S. Agrawal, “理解基因之间的相互作用算法参数，”在遗传算法的基础上 v, w. Banzhaf 和 C. Reeves。圣马蒂奥，加州：Morgan Kauffman, 1998 年,第 265-286 页。
- [6]K. Deb 和 R. B. Agrawal, “模拟二进制交叉连续搜索空间，”在复杂的系统，1995 年 4 月，卷。9,第 115-148 页。
- [7]C. M. Fonseca 和 P. J. Fleming, 《多目标优化的遗传算法：制定、讨论和推广》，第五届遗传算法国际会议记录，S. 佛雷斯特，爱德。圣马蒂奥，加州：摩根考夫曼，1993 年，第 416 - 423 页。
- [8]，“关于随机的性能评估和比较“多目标优化器”，从自然界中并行解决问题四、H-m.Voigt, W. Ebeling, I. Rechenberg, 和 H-P. SchwefelEDS。德国柏林：1996 年，《春天-威拉格》，P. 584 - 593。
- [9]，“多目标优化和多重约束处理有进化算法—第一部分：统一的公式，”这是我的荣幸系统，伙计，Cybern. a, 第二卷。1998 年 1 月 20 日，第 26 页至第 37 页。
- [10]，“多目标优化和多重约束处理有进化算法—第二部分：应用实例，”ieee 这是我的荣幸系统，伙计，Cybern. a, 第二卷。28,第 38-47 页，1998 年 1 月。
- [11]D. E. Goldberg、B. Korb 和 K. Deb, 《混乱的遗传算法：动机、分析和第一个结果》，在复杂的系统中，1989 年 9 月，卷。3、第 93-530 页。
- [12]G. Harik, “学习基因联系，有效的解决运用遗传算法时遇到了很大的困难。伊利诺斯

州的城市，城市，城市，城市 1997 年，第 97005 号报告。

[13]J. Horn、N. Nafploitis 和 D. E. Goldberg, “一个没有血缘关系的帕雷托人多目标优化算法,”在第一个关于进化计算的会议, Z. Michalewicz, 爱德。皮斯卡托维, nj: 艾埃出版社, 1994 年, 第 82-87 页。

[14]J. Knowles 和 D. Corne, “帕累托存档进化策略: a 多目标优化的新基线算法,”关于进化计算的 1999 年大会。-我不知道《新闻》, 1999 年, 第 98-105 页。

[15]F. Kursawe, “向量优化进化策略的变体,”在平行的问题解决自然, h. -p. Schwefel 和 R. Männer, EDS. 德国柏林: 斯普林格-威拉格, 1990, 第 193 - 197 页。

[16]C. Poloni, “用于多目标气动形状优化的混合 ga,”在工程和计算机科学的遗传算法, g. 冬天, J. Periaux, M. Galan 和 P. Cuesta. 纽约: 威利, 1997 年, 第 397-414 页。

[17]T. Ray, K. Tai 和 C. Seow, “多目标优化的进化算法”。最佳, 卷. 33, 第 3 号, 第 399-424 页, 2001 年。

[18]G. Rudolph, “部分有序集合下的进化搜索”。这是我的荣幸西施。/ls11, 大学。多特蒙德, 多特蒙德, 德国, 科技。代表。ci-67/99, 1999 年。

[19]J. D. Schaffer, “多目标优化评估向量《遗传算法》, 在首届国际遗传算法会议记录, J. J. Grefensttete, ed. 纽约, 希尔斯代尔: Lawrence Erlbaum, 1987 年, 第 93-100 页。

[20]N. Srinivas 和 K. Deb, “利用埃沃尔说: “我们的研究表明, 基因算法的排序是不受控制的。”计算, 卷. 2、不. 3, 第 221 页至第 248 页, 1995 年秋季。

[21]M. Tanaka, “基于 GA 的多细胞优化决策支持系统”。我的天。(单位: 千美元)系统, 人与网络-2, 1995 年, 第 1556-1561 页。

[22]D. Van Veldhuizen, “多目标进化算法: 分类、分析和新创新”。技术人员, 戴顿, 哦, 技术。报告附件/d/eng/99-01, 1999 年。

[23]D. Van Veldhuizen 和 G. Lamont, “多目标进化”算法研究: 一个历史与分析, “空军内部”。技术, 戴顿, 哦, 技术人员。1998 年 88-03 期代表。

[24]E. Zitzler, “多目标优化的进化算法: 《方法和应用》, 博士论文 Es 13398, 瑞士 1999 年, 瑞士苏黎世联邦理工学院。

[25]E. Zitzler, K. Deb 和 L. Thiele, “多目标进化算法的比较: 经验结果,”埃沃尔。计算, 卷. 《刑事诉讼法》第 8 条第 2 款。173-195, 2000 年夏季。

[26]E. Zitzler 和 L. Thiele, “使用进化算法的多目标优化——一个比较案例研究”, 并行问题从自然、v. A. E. Eiben、T. Bäck、M. Schoenauer 和 h. -p. Schwefel, 伊兹。德国

柏林：斯普林格-威拉格，1998，PP。292-301。



Kalyanmoy Deb (A'02) received the B.Tech degree in mechanical engineering from the Indian Institute of Technology, Kharagpur, India, 1985 and the M.S. and Ph.D. degrees in engineering mechanics from the University of Alabama, Tuscaloosa, in 1989 and 1991, respectively.

He is currently a Professor of Mechanical Engineering with the Indian Institute of Technology, Kanpur, India. He has authored or coauthored over 100 research papers in journals and conferences, a number of book chapters, and two books:

Multiobjective Optimization Using Evolutionary Algorithms (Chichester, U.K.: Wiley, 2001) and *Optimization for Engineering Design* (New Delhi, India: Prentice-Hall, 1995). His current research interests are in the field of evolutionary computation, particularly in the areas of multicriterion and real-parameter evolutionary algorithms.

Dr. Deb is an Associate Editor of IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and an Executive Council Member of the International Society on Genetic and Evolutionary Computation.



Amrit Pratap was born in Hyderabad, India, on August 27, 1979. He received the M.S. degree in mathematics and scientific computing from the Indian Institute of Technology, Kanpur, India, in 2001. He is working toward the Ph.D. degree in computer science at the California Institute of Technology, Pasadena, CA.

He was a member of the Kanpur Genetic Algorithms Laboratory. He is currently a Member of the Caltech Learning Systems Group. His current research interests include evolutionary computation,

machine learning, and neural networks.



Sameer Agarwal was born in Bulandshahr, India, on February 19, 1977. He received the M.S. degree in mathematics and scientific computing from the Indian Institute of Technology, Kanpur, India, in 2000. He is working toward the Ph.D. degree in computer science at University of California, San Diego.

He was a Member of the Kanpur Genetic Algorithms Laboratory. His research interests include evolutionary computation and learning both in humans as well as machines. He is currently developing learning methods for learning by imitation.



T. Meyarivan was born in Haldia, India, on November 23, 1977. He is working toward the M.S. degree in chemistry from Indian Institute of Technology, Kanpur, India.

He is a Member of the Kanpur Genetic Algorithms Laboratory. His current research interests include evolutionary computation and its applications to biology and various fields in chemistry.