用粒子群算法解决约束非线性优化问题

摘要：

本文提出一个粒子群优化算法（PSO）去解决带约束的非线性优化问题。在PSO中，通过学习当前的最佳粒子及其自身的记忆，将潜在的解决方案称为“微粒”在问题空间中“流动”。本文采用保留可行性策略来处理约束。PSO从一组开始可行性解决方案和可行性函数用于检查新探索的解决方案是否满足所有约束条件。所有粒子仅将那些可行解决方案保留在其内存中。对11个测试案例进行了测试，表明PSO是解决大多数问题的有效且通用的解决方案 具有非线性不等式约束的非线性优化问题。

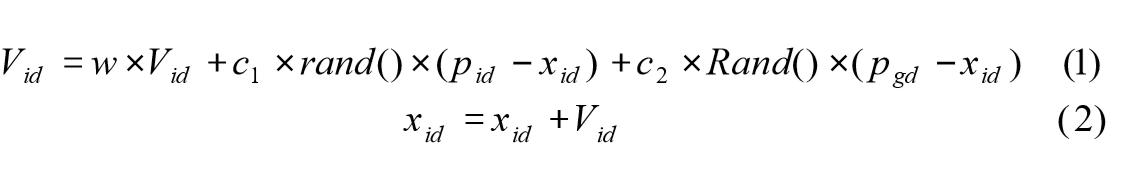
1. 简介

PSO是Kennedy和Eberhart共同开发的一种进化计算技术，它使用了常见的进化计算技术：1使用大量随机解进行初始化;

2.通过更新世代来寻找最优值;

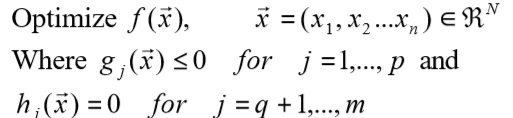
3.更好的进化是基于先前的子代， 在PSO中，通过跟随当前的最佳粒子，使称为粒子的潜在解决方案“流过”问题空间。

粒子的更新通过以下方程式完成。Eq(1)根据先前的速度（Vid）计算每个粒子（势解）的新速度，该粒子到目前为止已达到最佳适应度的位置（ Pid或pBest），以及总体（目前在算法的本地版本中为本地邻居）位置（Pgd，对于全局版本而言为gBest或对于本地版本而言为Pid，lBest）的位置（迄今为止达到最佳适应性）。Eq(2)更新了每个粒子在解超空间中的位置，两个随机数c1和c2是独立生成的，惯性权重w的使用在许多应用中提供了改进的性能。



事实证明，粒子群算法对许多优化问题非常有效。在这一领域已经做了很多工作[4]。 但是，约束优化问题仍然是粒子群优化的新领域。

一般约束非线性优化问题(CNOPs)定义如下:



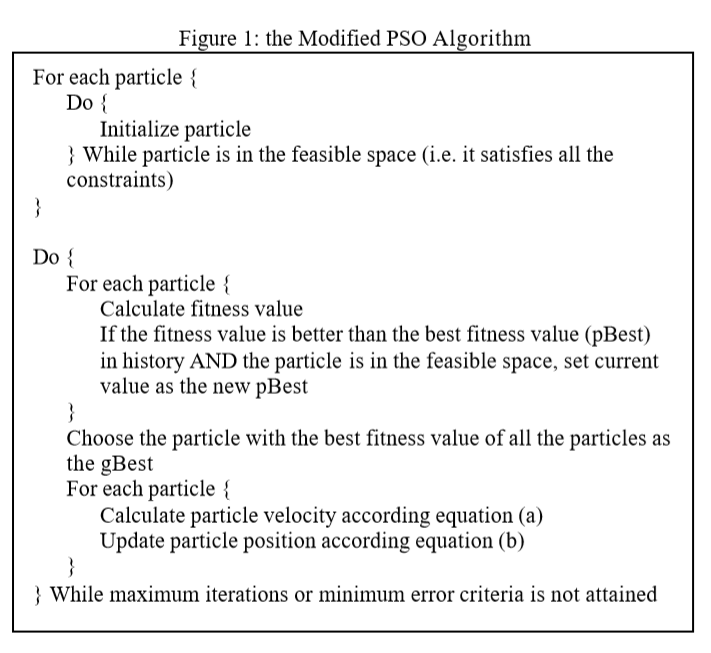
它们由三个基本组件组成:一组变量、要优化的适应度函数(最小化、或最大化)和一组指定变量可行空间的约束。目标是在满足约束条件下，找出优化适应度函数的变量值。

1. 背景

近几十年来，人们对CNOPs进行了大量的研究。由于非线性优化的复杂性和不可预测性，一般的确定性求解是不可能的。这为进化算法提供了一个机会。近年来，一些进化算法存在优化问题。Michalewicz[6]概述了这些算法。本文综述了几种演化计算方法，并给出了一组有约束的数值优化试验用例。

约束优化过程的关键是约束的处理。提出了许多处理约束的方法。Koziel等人[5]将其分为四类:基于保持解的易用性的方法;基于惩罚项的方法;明确区分可行解和不可行解的方法;和其他混合方法。

对于CNOPs，需要修改原始的PSO方法来处理约束。可以采用上述约束处理方法中的一些思想。最直接的方法是基于保留解的可行性的方法。为了在可行空间中找到最优解，每个粒子搜索整个空间，但只跟踪可行解，为了加速这个过程，所有粒子都初始化为可行解。下面是算法。



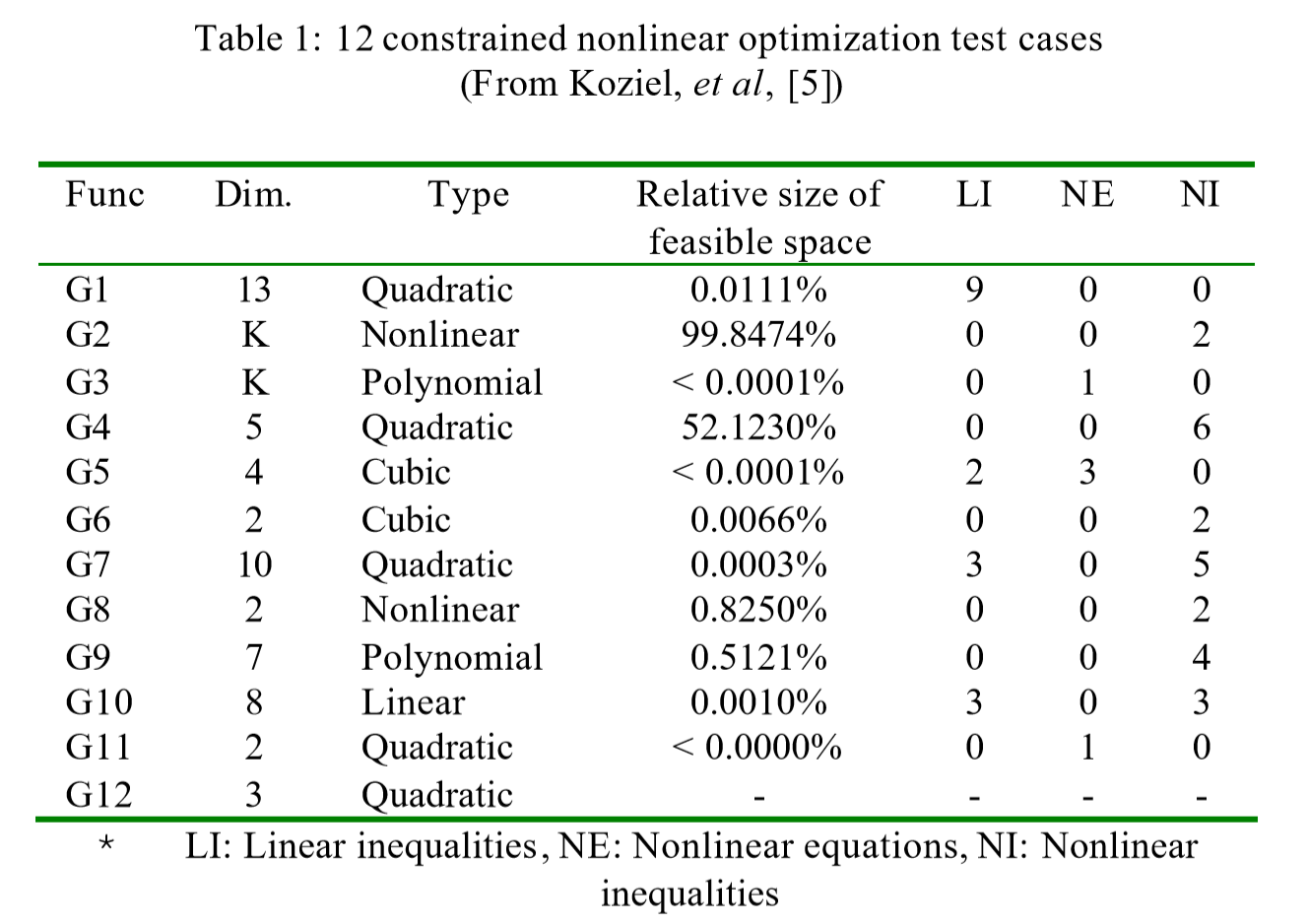
与原始PSO相比，有两个修改。

1. 在初始化过程中，重复初始化所有的粒子，直到它们满足所有的约束。
2. 在计算pBest和gBest值时，只计算可行空间中的位置。

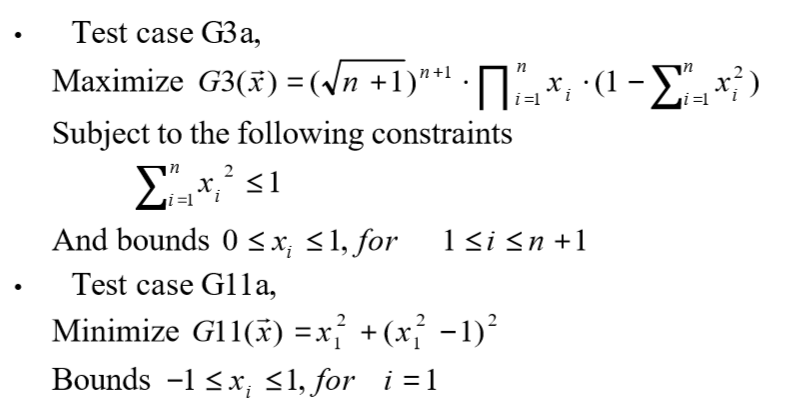
上述算法是全局版本。对于局部版本，算法只有一个不同之处:每个粒子不是找到最好的，而是找到一个最优(lBest)来更新新速度。

1. 实验设置

对12个约束数值优化问题进行了实验验证。它们是由Michalewicz和Schoenauer[6]提出的。这些测试用例包括具有不同约束类型的目标函数[5,6]。详细功能信息请参考文献。



在表格1中。有三个处理非线性方程约束的测试用例：G3、G5和G11。在PSO方法中，所有随机初始化的解都需要位于可行空间中。对于这些方程约束条件，随机生成一组可行解几乎是不可能的。有几种技术可以消除这些方程约束。一是采用直接替换的方法来消除约束，具体如下:



测试用例G12是一个特殊情况，其中在搜索空间中嵌入了125个不相交的可行空间球体。

为了与其他人的结果进行比较，每个测试用例都进行了两种类型的实验：

实验＃1：执行了20次运行。 对于每次运行，最大代数设置为T =200。

实验2：执行了20次运行。 对于每次运行，最大代数设置为T = 500。

1. 参数选择

PSO的粒子群大小通常在10到40之间。Carlisle[7]比较了不同的粒子群大小，并建议30为合适的选择。在这里，粒子群大小为20。粒子群较小的原因是它大大减少了计算时间。 这是因为在初始化期间，所有粒子都必须在可行的空间内。随机初始化的粒子并不总是在可行的空间中。因此，如果种群太大，初始化可能会花费更长的时间。但是，对于某些复杂的情况，首选较大的粒子群规模。

在PSO中，不需要调整很多参数。仅需要注意以下几个参数：最大速度Vmax，惯性权重w，认知学习率c1和社交学习率c2。参数设置与以前一样[8;9]。惯性权重为[0.5 +（Rnd / 2.0）]。学习率是1.49445。 最大速度Vmax设置为粒子的动态范围。

1. 结果

下表总结了11个带有非线性不平等约束的测试用例的结果。所有的问题都变成了最小优化问题。因此，这里显示的所有最优值都是最小值。首先，需要注意的是，PSO能够成功地找到除测试用例G5之外的所有情况的最优解，而测试用例G5是由于方程的约束没有得到求解。

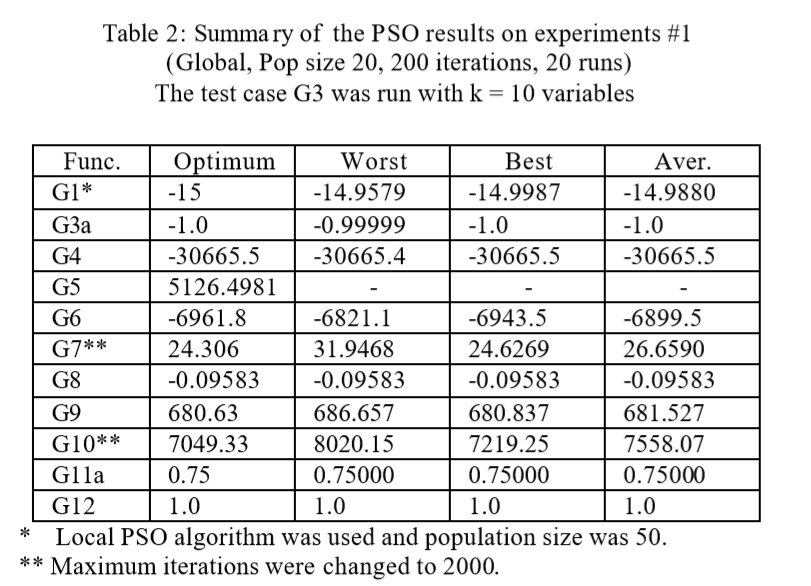
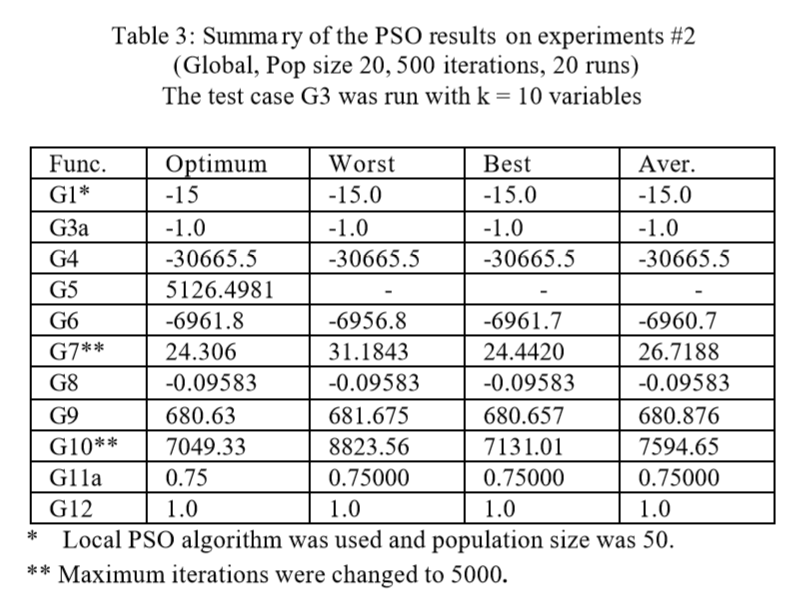
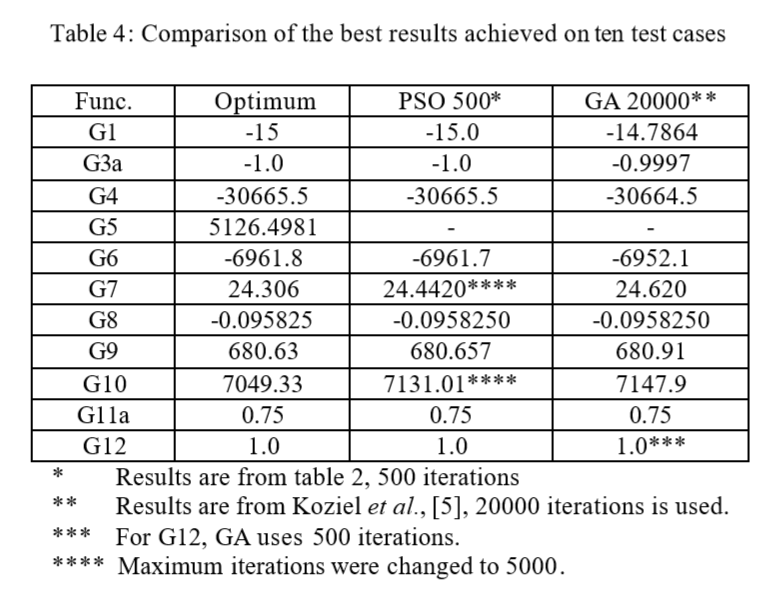


表2显示，在剩下的11个测试用例中，有10个测试用例，包括两个修改过的用例G3a和G1 la, PSO在200次迭代中成功地找到了最优或接近最优。表3为第二次实验结果。与其他结果相比，这要比之前报道的其他结果快得多[5,10]。

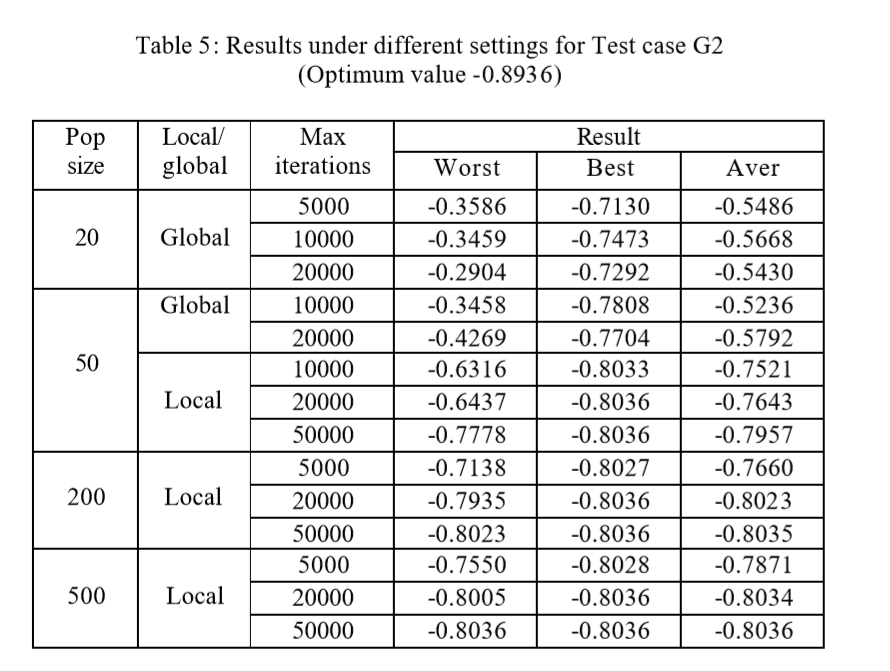


除了速度，粒子群优化算法还发现了比上述算法更好的结果。表4是20次运行的最佳测试结果的逐行比较。结果表明，PSO在迭代次数少的情况下得到了较好的或相似的结果。



除测试用例G1外，其余均采用全局PSO算法。在G1用例中，全局粒子群优化算法有时会陷入局部极小值。这个问题可以通过使用PSO算法的局部版本来解决。粒子群规模也改为50。然后PSO毫无例外地找到了全局最优解。对于测试用例G7和G10，使用了更高的最大迭代数。最大迭代被分别更改为2000和5000。粒子群优化算法在较短的时间内得到了较好的结果。

测试用例G2是PSO没有使用上述设置找到最佳区域的唯一情况。尝试了不同的种群大小、最大迭代次数和局部版本的粒子群优化算法。具有较大种群规模的局部粒子群优化算法成功地找到了全局最优解。结果如表5所示。



1. 讨论

已知，不同的约束处理技术对不同的CNOP情况提供了不同的质量结果从以上结果可以看出，在所有CNOP情况下，PSO算法对全局最优区域的定位是一致的。由于进化算法的起源是随机的，方程约束很难处理。在可行空间中几乎不可能找到一组初始解。这也适用于那些可行空间极小的问题。

与其他方法相比，PSO具有以下优点:

1. 算法简单，需要调整的参数不多。

2. 该算法功能强大，对于上述基准函数PSO速度更快，上述结果也表明，该算法可以处理多种约束条件下的优化问题。

3.目标和约束没有预定义的限制;它不需要对目标和约束进行预处理。

在实际问题中可以使用两个版本的粒子群优化算法。局部PSO算法是更准确的结果，而全局版本是更快的首选。如果结果不令人满意，可以尝试更高的迭代次数和更高的种群大小。

本文提出了一种用于约束优化的粒子群优化算法。结果表明，粒子群优化算法是一种有效、通用的求解约束参数优化问题的方法。本文仅代表了用粒子群算法求解约束参数优化问题研究的第一步。为了在实际应用中发挥作用，需要证明粒子群解决更复杂约束优化问题的能力。