

搜索问题和搜索方法的统一表述

第二个实验使用了 ACO 算法, GA 算法以及 Hopfield 网络实现对 TSP 问题的求解。对这些方法有了一定的体会, 包括搜索的效率, 收敛的速度, 得到的解的优劣程度等等。

结合上课的内容与本次实验的结果总结, 对搜索问题和搜索方法进行阐述。

1.搜索问题和搜索方法的表述

搜索问题是在一个限定的解空间中去寻找一个目标或者在多项式时间内寻找到一个近似最优解的过程。

搜索问题的解空间有两种表示的方法:

- 状态空间表示法。将问题求解所涉及的每个可能的步骤表示成一个状态。全部状态以及状态空间之间的所有转换构成一个以图的形式来表示的状态空间。问题的求解过程是在状态空间中搜索一条最优的或者可行的从初始状态到目标状态的路径的过程。
- 与或图表示法。将问题归约, 从原始问题到简单问题再到本原问题的所有可能的规约表示成一个“与/或图”。问题的求解过程是在与或图中搜索一个将原始问题转化为简单问题再变换为本原问题的、最优的或可行的归约步骤的过程。

搜索方法:

一般而言, 搜索方法所指的就是图搜索方法, 图搜索方法中包含了树搜索方法等。搜索方法顾名思义就是一种在解空间上搜索到所需解的方式。

图搜索一般采用 Open 表和 Closed 表。

搜索方法可以分为两大类: 盲目图搜索和启发式图搜索。

在解决实际的搜索问题时, 需要先确定问题的状态, 然后将状态形成解空间, 使用具体的搜索方法在解空间中搜索, 得到需要的最优解。这两者是一个搜索问题的解决的基本组成部分, 正是由于对于实际问题而言, 无法表示出完整的“状态空间图”或“与或图”, 所以需要搜索方法来完成盲目或者启发式的对解空间的搜索。

2.搜索算法的 Exploration 和 Exploitation 能力

Exploration 和 Exploitation 本身就是一种矛盾体。Exploration 体现出算法的搜索新空间的能力, 而 Exploitation 则体现了算法求精的能力。这两项指标是评价随机优化算法的标准。

如果把寻找解的问题形象地看作是勘探矿物的话, 这两个概念就非常明晰了。Exploration 说的是探索有矿物的地点的能力, 而 Exploitation 则是一种已经知道了某个地方有矿物, 而去开采, 深度挖掘。

抽象地说, Exploration 是在整个搜索空间中搜索解的能力(全局搜索能力), 该属性越强, 优化算法就不会局限于局部的搜索, 更加容易发现全局的最优解, 与此同时, 搜索到解的效率也会降低。Exploitation 是在当前解的附近的搜索能力(局部搜索能力), 与前者相反, 其更容易在当前解所在解空间的位置的附近快速地找到一个局部最优解, 体现了算法的开发优化解的能力。搜索解的效率

提高，伴随着的是容易陷于局部最优。

下面，结合本次的实验的两个优化算法，比较他们 Exploration 和 Exploitation。

GA 算法解决 TSP 问题：

GA 算法最大的特点就是并行计算的能力。可以同时多个解迭代计算，但是伴随的问题就是其 Exploration 能力。迭代一次，进行重组操作和使用轮盘赌算法去选择比父代更优的个体，就是在当前最优解的附近寻找到更优的解。而变异操作则是在当前解上去任意跳到新的解。实际问题是变异操作其实收到的效果是微弱的，除非变异的频率需要加大，但是这样就会使收敛的速度下降。并且 GA 算法常常会过早收敛。

在 ACO 算法中。信息素浓度体现了算法的 Exploitation，这是在局部较优解的附近去探寻更优解，信息素的浓度提供了其局部探索的方向。而在选择下一个城市的时候使用的轮盘赌算法，给出了选择的概率，在一定程度上缓解了陷于局部的困扰，体现了 Exploration 的能力。

3.对于优化算法的优化:

对于提高 Exploration:

- 可以多次运行得到解，在得到的解集中寻找到最好的解。这需要人力物力的消耗，效率过低。
- 扩大算法的 Exploration 的属性。
在 GA 算法中，可以不断加大变异的频率；在 ACO 算法中可以不断加大其随机的属性，比如可以适当修改轮盘赌的规则，让其全局的搜索能力加强。
- 可以空大对解空间的覆盖，加大初始解的规模。

对于提高 Exploitation

- 在 GA 算法中就可以加大重组的比重，尽量加大父代与子代的相似度，ACO 算法则可以使用轮盘赌中比例较大的比例，这也形象地称为减小学习步长

在实验过程中，也提到了在 ACO 算法中可以使得蚂蚁自身的信息素浓度在运行过程中也发生变化，这可以增加 Exploration 的能力；在 GA 算法中随着迭代的次数增多，从而减小突变的概率，这可以增加 Exploitation 的能力，这都是动态的改变学习的补偿。

在实际搜索时，还有一些别的算法可以帮助权衡两者，比如模拟退火算法等。

在实际的问题中，两者是需要综合两者考虑的；一个算法的优化能力，两者都需要考虑和权衡。