

论贪心算法在图论中的应用

汪莹 / 南京邮电大学 计算机学院

摘要: 本文介绍了贪心算法的基本概念和解题思想, 并通过两个典型的实例, 说明了在图论中贪心算法的具体的应用。

关键词: 贪心算法; 最优子结构性性质; 贪心准则

贪心算法是用来求解某些问题最优解的一种有效算法, 它可以简单迅速的解决很多实际应用的, 在图论领域也有着非常广泛的应用。

1 贪心算法概述

1.1 贪心算法定义。贪心算法是一种经过改进的分级处理的方法。贪心算法在进行决策的时候总会做出对目前情况最好的选择。和动态规划算法不同的是, 贪心算法并不考虑问题的整体最优, 而只是选择某种意义上的局部最优。所以, 贪心方法未必能针对所有问题求得整体的最优解, 但是对于许多问题而言, 它却能得到整体最优解。在某些情况下, 贪心算法虽然得不到整体最优解, 却可以得到最优解的很好的近似。

1.2 贪心算法的基本要素。对于一个具体问题, 怎样才能知道能否可以用贪心算法求解并得到最优解呢? 在实际应用中人们总结出了两个重要的性质:

(1) 贪心选择性质; (2) 最优子结构性性质。

许多可以利用贪心方法求解的问题一般都具备这两个性质。

1.3 贪心算法的求解步骤。贪心算法求解问题首先要确定一个贪心准则, 每一步都是按这个准则选取优化方案。按照这种度量标准对此问题的 n 个输入排序, 并按序一次输入一个量。如果这个输入和当前已构成在这种量度意义下的部分最优解加在一起不能产生一个可行解, 则不把此输入加到这部分解中。因此贪心准则的选取直接关系到可行解能否达到最优。

2 贪心算法求解单源点最短路径问题

图论中有一个非常典型的问题: 单源点最短路径问题。问题描述如下: 对于有向带权图 $G = (V, E)$, 其中每条边上的权值是一个非负实数。选定 V 中的一个顶点, 称之为源点。这个问题求解的是从源点到图 G 中其他所有顶点的最短路径的长度。所谓的长度是指经过的各边上的权值总和。

Dijkstra算法正是利用贪心算法求解此问题, 用迄今已生成的所有路径长度之和为最小作为贪心准则, 为此, 每一条单独的路径都必须具有最小长度。此算法的实现步骤为: 将顶点集合分为 S 和 $V-S$ 这两个子集。子集 S 中均为已求得最短路径的顶点。初始时, 集合 S 中只有源点, 接下来不断做贪心选择来扩充此集合。取 G 中的一个顶点 v , 把从源点到 v 点并且只经过 S 中顶点的路径称为从源点

到顶点 v 的特殊路径, 设置数组 $dist$ 记录每个顶点当前所对应的最短特殊路径的长度。Dijkstra算法每次都从子集 $V-S$ 中取出特殊路径长度为最短的顶点, 并把该顶点添加至子集 S , 随后对数组 $dist$ 进行修改。一旦子集 S 等于顶点集 V , 算法结束。数组 $dist$ 中记录的就是源点到图中其他各顶点的最短路径的长度。例如, 下图1中的有向带权图, 采用Dijkstra算法, 从源点1到图中剩余各顶点间最短路径的计算过程如下表1所示:

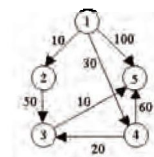


图1

表1 Dijkstra算法的求解过程

迭代	S	u	dist[2]	dist[3]	dist[4]	dist[5]
初始	{1}	-	10	maxint	30	100
1	{1,2}	2	10	60	30	100
2	{1,2,4}	4	10	50	30	90
3	{1,2,4,3}	3	10	50	30	60
4	{1,2,4,3,5}	5	10	50	30	60

3 贪心算法求解最小生成树问题

最小生成树是非常具有实际应用价值的图论问题, 具体描述如下: 已知无向连通带权图 $G = (V, E)$ 。E中每条边 (u,v) 的权值设为 $c[u][v]$ 。求图 G 的生成树 G' , 使得 G' 上各边权值的总和是所有生成树中各边权值总和最小的。这个问题被称为最小生成树问题。最小生成树的求解方法有多种, 其中应用比较广泛的几种算法, 如Prim算法和Kruskal算法都是采用了贪心算法的策略。

3.1 Prim算法

Prim算法的基本思想为: 首先选择图中任意顶点 u , 并将 u 置于顶点集合的子集 S 中。如果 S 是顶点集合 V 的真子集, 就继续选择顶点 j 添加到子集 S 中, 这里的 j 必须满足条件 $i \in S, j \in V-S$, 且 $c[i][j]$ 是权值最小的边。按照同样的步骤不断进行贪心选择, 直到子集 $S=V$ 为止。此时, 选取到的所有的边就构成了图 G 的一棵最小生成树。例如, 对于图2中的无向连通带权图, 按照Prim算法求最小生成树的过程如图3所示。

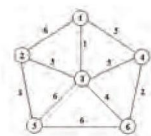


图2

电容触摸模块的功能实现和键盘矩阵原理有些类似,其基本原理是通过多个电容感应传感触点组成一个感应矩阵区,当电容触摸模块有效时,电容感应专用的MCU会定期对每个传感触点上的电容量进行扫描,如果有检测到某此传感触点上的电容值变化超过设定的阈值时,则立即启动所有的传感触点的快速扫描,将所有的传感触点的变化值记录下来,同时结合硬件上传感触点的矩阵分布,通过对每一个传感触点电容值的变化分析,确定电容变化点在模块上的分布状况,从而分析用户在模块上的操控轨迹;最终通过分析用户操控的轨迹来判断用户最终的操控意图。基于这种原理,在一个很小的触摸区域内,可以通过简单的算法实现用户的以下基本操作:上、下、左、右滑动,同时还可能识别用户滑动的速度变化及快慢。

三轴加速度及三轴陀螺仪模块主要用于实现空中鼠标功能。所谓的空中鼠标就是用户通过在空中移动遥控器来操控鼠标的功能。从原理上来看,其实现的最终目的是需要将用户在空间的三维移动转变为二维移动。从实际经验来看,要实现较好的操控体验,需要将三轴加速度传感器和三轴陀螺仪传感器结合起来使用,因为在一个三维的

移动过程中,需要考虑的因素主要有:方向变化、速度变化、以及角度的变化。简单通用的算法中一般可以通过加速度传感器的数据变化可以分析用户操作时的方向变化和速度变化,在算法中可能将用户操控前的状态设定为原始操控状态,将此时的三维空间通过映射产生的二维平面设定为原始二维平面,后续的操作都可以参考此平面。这样就可以将用户在三维上的移动转变为在此二维平面上的移动。又因为用户在三维移动过程中会出现角度的变化,可以通过相应的算法将角度变化对方向变化及速度变化的影响进行修正,这样用户在三维空间移动映射在二维平面上的轨迹就比较接近用户期望的移动轨迹了。但是如果想要得到一种更好的体验效果,则需要不停地分析用户的移动轨迹,动态地更新参考平面,这样才能使用户每一次的移动更好地在二维平面上进行反馈。

RF发射模块主要是将相应的数据流通过RF方式发送给相应的控制设备。随着控制传输的数据量增大,原有的红外传输方式已不能满足要求,因此在智能遥控器其主要采用RF传输方式,而其传输协议主要有:非标私有协议、蓝牙、Zigbee、WIFI等。

参考文献:

[1] 李蕾,刘卫东.智能遥控器中的MEMS惯性传感器数据处理[J].电脑知识与技术,2011,07(19):4639.

批注:

①利用化学试剂的溶液,借助于化学或电化学作用显示金属的组织。

作者简介:王光法(1974.12-),男,安徽庐江人,讲师,本科,主要研究方向:计算机基础教学、人机交互。

作者单位:江苏省连云港财经高等职业技术学校,江苏连云港 222000

《《《《《上接第309页



图3

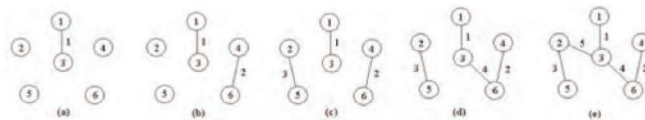


图4

3.2 Kruskal算法

Kruskal算法构造最小生成树的过程相比Prim算法更加简单明了。首先将无向连通带权图G的n个顶点作为n个孤立的连通分支。并按照权值给所有的边从小到大进行排序。将第一条边加入到连通分支,然后按权值递增的顺序依次查看剩下的每一条边,如果这条边加入构成回路就抛弃,继续考察下一条边,反之加入到连通分支中。一旦加入边的条数等于n-1,则算法结束。剩下唯一的连通分支就是要求取的最小生成树。

例如,对上图2,按Kruskal算法求得最小生成树的过程如下图4所示。

比较上述两种方法可以看出,虽然都是利用贪心算法解决,但是由于选择的贪心准则不一样,因次求解步骤和最小生成树的选边过程都是不一样的。

4 总结

贪心算法也存在着缺点,比如应用范畴比较狭窄而且有一些极难证明,但是针对图论的部分问题,比如上述的两个典型问题,贪心算法不但能够正确的求解,而且与其他算法相比,更是体现了其高效的优点。当然,图论中还有许多未解的问题,这些问题能否用贪心算法求解,或者能否找到更好的贪心准则提高求解的效率则需要更加深入的研究和发现。

参考文献:

[1] 余祥宣,崔国华,邹海明.计算机算法基础(第三版)[M].华中科技大学出版社,2006.

[2] 王晓东.算法设计与分析(第2版)[M].清华大学出版社,2008.

[4] 杨克昌.计算机常用算法与程序设计案例教程[M].清华大学出版社,2011.

[5] 王桂平,王衍,任嘉辰.图论算法理论、实现及应用[M].北京大学出版社,2011.

作者单位:南京邮电大学 计算机学院,南京 210023