

# 最短路径问题及其解法研究

周先曙

(中国人民解放军 75742 部队, 广东 广州 510515)

**摘要:**最短路径问题是在给定的网络图中寻找出一条从起始点到目标点之间的最短路径。该文分别从动态规划、Dijkstra、A\* 算法、遗传算法这四种算法设计方法入手, 概述了各种设计方法的原理, 提出了求解最短路径的算法思想, 并对算法进行分析, 提出了改进方法。

**关键词:**最短路径; 动态规划; Dijkstra 算法; A\* 算法; 遗传算法

**中图分类号:** TP312    **文献标识码:** A    **文章编号:** 1009-3044(2010)06-1403-03

## Shortest Path Problem and its Solution Methods Study

ZHOU Xian-shu

(People's Liberation Army 75742 Units, Guangzhou 510515, China)

**Abstract:** Shortest path problem is to find a shortest path from the start point to the end point in a given net graph. This paper will introduce four algorithm design methods, which are Dynamic programming, Dijkstra algorithm, A-star algorithm, Genetic algorithm, summarize their basic tenets, give the solving algorithm thought to the shortest path problem, analyse the algorithms and put forward the improving methods.

**Key words:** shortest path; dynamic programming; Dijkstra algorithm; A-star algorithm; genetic algorithm

最短路径问题是在给定的网络图中寻找出一条从起始点到目标点之间最短路径。最短路径算法不仅在 GIS 的交通路线导航、路径分析领域应用广泛, 在解决路径搜索相关的应用中也十分普遍, 包括网络路由算法、机器人探路、人工智能、游戏设计等等。本文分别从动态规划、Dijkstra、A\* 算法、遗传算法这四种算法设计方法入手, 概述了各种设计方法的原理, 提出了求解最短路径的算法思想, 并对算法进行分析, 提出了改进方法。

## 1 最短路径的搜索算法概要

### 1.1 动态规划算法

动态规划算法的基本思想是将待求解问题分解成若干个子问题, 先求解子问题, 然后从这些子问题的解得到原问题的解。动态规划算法使用于解最优化问题。通常可以按以下步骤设计动态规划算法:

- 1) 找出最优解的性质, 并刻画其结构特征;
- 2) 递归地定义最优解;
- 3) 以自底向上的方式计算出最优解;
- 4) 根据计算最优值时得到的信息, 构造最优解。

步骤 1)~3) 是动态规划算法的基本步骤。在只要求最优值的情形, 步骤 4) 可以省去。若要求问题的最优解, 则必须执行步骤 4), 此时, 在步骤 3) 中计算最优解时, 通常需记录更多的信息, 以便在步骤 4) 中, 根据所记录的信息, 快速构造出最优解。

### 1.2 Dijkstra 算法

Dijkstra 算法是 Dijkstra 提出的一个按路径长度递增的顺序逐步产生最短路径的算法。该算法的基本思想是: 设置两个定点的集合 T 和 S, 集合 S 中存放已找到最短路径的定点, 集合 T 中存放当前还未找到的最短路径的定点。初始状态时, 集合 S 中只包含原点  $v_0$ , 然后不断从集合 T 中选取到定点  $v_0$  路径长度最短的顶点  $u$  加入集合 S, 集合 S 中每加入一个新的顶点  $u$ , 都要修改定点  $v_0$  到集合 T 中剩余顶点的最短路径长度值, 集合 T 中个顶点新的最短路径长度值为原来的最短路径长度值与定点  $u$  的最短路径长度值加上  $u$  到该顶点的路径长度值中的较小值。次过程不断重复, 直到集合 T 的顶点全部加入到集合 S 为止。

### 1.3 A\* 算法

A\* 算法则是人工智能领域的一种图搜索策略, 采用了启发式函数对搜索过程中产生的分支进行评估, 以选择最佳的分支进行搜索。其更一般的引入了一个估价函数  $f(n)$ , 其定义为  $f(n) = g(n) + h(n)$ 。其中  $g(n)$  为到达当前节点的耗费, 而  $h(n)$  表示对从当前节点到达目标节点的耗费的估计, 要求评估函数满足  $h(n) \leq h^*(n)$ 。其必须满足两个条件:

- 1)  $h(n)$  必须小于等于实际的从当前节点到达目标节点的最小耗费  $h^*$ 。
- 2)  $f(n)$  必须保持单调递增。

### 1.4 遗传算法

遗传算法是一类借鉴生物界自然选择和自然遗传机制的随机化的搜索算法, 由美国 J.Holland 教授提出, 其主要特点是群体搜

收稿日期: 2010-01-04

作者简介: 周先曙 (1984-), 湖南汨罗人, 信息工程大学电子技术学院广州训练大队训练处助教, 研究方向为信息安全。

本栏目责任编辑: 谢媛媛

万方数据

索策略、群体中个体之间的信息交换和搜索不依赖于梯度信息。因此它尤其适用于处理传统搜索方法难于解决的复杂和非线性问题。遗传算法是一种群体型操作,该操作以群体中所有个体为对象。选择、交叉和变异是遗传算法的三个主要算子,他们构成了遗传算法的主要操作,使遗传算法具有了其它传统方法所没有的特性。

2 最短路径问题的提出

如图 1,给定一个运输网络,两点之间连线上的数字表示两点的距离。求一条从 A 到 E 的运输线路,使总距离为最短。

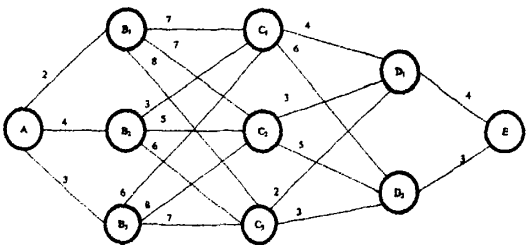


图 1

3 求解最短路径问题

3.1 利用动态规划解最短路径问题

3.1.1 算法实现

图 1 中,过程的始点是 A,终点是 E,过程的实际行进方向是由 A 经第 1,2,3,4 阶段到达终点 E。

在第 4 阶段有两个初始状态 D1 和 D2。那么,从 A-E 的全过程最短路径,在第 4 阶段究竟经过 D1、D2 中的哪一个呢?目前,我们还不得而知,因此只能各种可能都考虑;若全过程最短路径经过 D1,,则有  $f(D_1)=4$ ;若全过程最短路径经过 D2,则有  $f(D_2)=3$ 。

第 3 阶段假设全过程最短路径在第 3 阶段经过 C1 点,则有:

若由  $C1-D1-E$ ,则有  $d_3(C_1,D_1)+f_4(D_1)=4+4=8$ ;

若由  $C1-D2-E$ ,则有  $d_3(C_1,D_2)+f_4(D_2)=6+3=9$ ;因此,由 C1 到 E 的最短路径是由  $C1-D1-E$ ,最短距离是  $f_3(C_1) = 8$ 。

假设全过程最短路径在第 3 阶段经过 C2 点,则有:

$f_3(C_2)=\min\{[d_3(C_2,D_1)+f_4(D_1)],[d_3(C_2,D_2)+f_4(D_2)]\}=7$ ;因此,由 C2 到 E 的最短路径是由  $C2-D1-E$ ,最短距离是  $f_3(C_2)=7$ 。

同理由 C3 到 E 的最短路径是由  $C3-D1-E$  和  $C3-D2-E$ ,最短距离是  $f_3(C_3)=6$ 。

同理,可得到第 2 阶段由 B 到 E 的最短路径有三条;B1 到 E;B1-C2-D1-E 或 B1-C3-D1-E,最短距离都是 14;B2 到 E;B2-C1-D1-E,最短距离是 11;B3 到 E;B3-C3-D1-E,最短距离都是 13。

第一阶段,计算如下:

$f_1(A)=\min\{[d_1(A,B_1)+f_2(B_1)],[d_1(A,B_2)+f_2(B_2)],[d_1(A,B_3)+f_2(B_3)]\}=15$

因此,由 A-E 的全过程最短路径是 A-B2-C1-D1-E,最短距离是 15。

3.1.2 算法分析及改进

利用动态规划求解最短路径问题的复杂度为  $O(\min\{nc,2^n\})$ 。动态规划主要是求解最优决策序列,当最优决策序列中包含最优决策子序列时,可建立动态规划递归方程,它可以帮助高效地解决问题。用动态规划方法不仅能得到全局最优结果,还能得到一族最优结果。有了这一族最优结果,便于分析、比较不同的结果,且如果由于某种原因偏离了原始最优轨迹,也可以很方便地找出余下阶段的新的最优子策略。

3.2 利用 Dijkstra 算法解最短路径问题

3.2.1 算法实现

将图 1 采用邻接矩阵来存储,设为有向网  $G=(V,E)$ ;若邻接矩阵向 C,并规定:

$$C[i][j]=\begin{cases} w_{ij} \\ 0 \\ \infty \end{cases}$$

为  $w_{ij}$  表示顶点 i 和顶点 j 之间有直接边,且权值向  $w_{ij}$ ,为 0 表示  $i=j$ ,同一个顶点。为  $\infty$  表示顶点 i 和顶点 j 没有边。

设置一个一维数组 S,用来标记已找到的最短路径的顶点,并规定:

$S[i]=0$  表示未找到源点到顶点  $v_i$  的最短路径;

$S[i]=1$  表示已找到源点到顶点  $v_i$  的最短路径;

初始状态时,  $S[0]=1$ ,对于其余的顶点  $v_i(v_i \neq v_0)$ ,有  $S[i]=0$ ;设置另一个一维数组 D,用来存放源点到其余各顶点的当前最短路径长度。初始状态时,  $D[i]=C[ORG(v_0)][i]$ (其中  $ORG(v_0)$  表示源点  $v_0$  在存储结构中的位置号),即一维数组 D 各分量的值为邻接矩阵 C 中  $ORG(v_0)$  行上各元素的值。

首先利用 D 数组中各分量的值,选取当前具有最短路径的顶点 u,使得  $D[u]=\min_{j \in V} \{D[j]\}$ ,

然后将顶点 u 加入集合 S 中,即令  $S(u)=1$ ,同时对于所有的  $S[i]=0$  的顶点  $v_i$ ,判断  $D[i]$  是否小于  $D[u]+C[u][i]$ ,使得

$D[i]=\min\{D[i],D[u]+C[u][i]\}$

上述过程重复执行 n-1 后,就得到了按路径长度递增的顺序求源点到其余各顶点的最短路径值。

3.2.2 算法分析及改进

Dijkstra 算法的成功率是最高的,因为它每次必能搜索到最优路径。另一方面,有利必有弊,如果要找到最优路径必然要在速度上做出牺牲。这导致 Dijkstra 算法的搜索速度是最慢的。Dijkstra 算法求一个点到其他所有点的最短路径复杂度为  $O(n^2)$ ,求一个点到其他所有点最短路径等效于求一个点到另一个点的最短路径 n 个点之间互相的最短路径复杂度为  $O(n^3)$ 。

### 3.3 A\* 算法解最短路径问题

#### 3.3.1 算法实现

A\* 算法的基本思想是:

Procedure A- star

Begin

1) 把源节点放入队列 Q;

2) For(i=2,i<maxn,i++)

Begin

If 队列 Q 为空 then 退出; 删除队列中最小代价结点 current

If current==goat then 结束; 找到最优;

If current 有子女, 源结点到本结点权值更小, 即有更新, 且该结点不在队列, 则插入;

End For

输出路费和路径

End.

#### 3.3.2 算法的分析与改进

空间复杂性: A\* 算法生成的总节点数为 N, 解的深度为 d, 那么 b\* 就是尝试为 d 的一致搜索树为了包括 N+1 个节点所必需的分支因子。因此,  $N+1=1+b^1+(b^2)+\dots+(b^d)$ , 从中可看出空间复杂度高。

时间复杂性: 充分利用问题内在信息, 启发函数设计的好, 可以极大降低扩展的结点, 从而在较小的时间内即可完成最佳路径搜索。

如果 h(n) 是可采纳的, 那 A\* 算法是完备的也是最优的。

### 3.4 遗传算法解最短路径问题

#### 3.4.1 算法实现

1) 编码方案: 设  $a_{ij}$  表示第 i 个结点到第 j 个结点的路径信息,  $1 \leq i \leq N, 1 \leq j \leq N$ , (其中 i, j 为整数, N 为网络中的结点的个数)。若  $a_{ij}$  为零表示两个结点间无直接路径存在; 不为零则表示这两点间存在直接路径为  $a_{ij}$ 。

2) 初始群体的生成: 将个体初始化成零个体, 即置个体点的集合信息为:  $a_i=0, a_i=0(0 \leq i \leq N-1), a_i=N^{(i=0)}$ , 对每个个体作同样操作。然后对整个群体进行初始化。

3) 适应度函数的设计: 根据个体所指示的路径信息, 判断这条路径中没两个点间的距离大小, 若两点间存在路径且没有到达最后一个点, 则将适应值的值加上当前两点间的路径长度, 若不存在连线, 则将适应值加上惩罚值, 最后保存该个体的适应值, 以便在选择时进行优先选择。对每一个个体进行此操作, 直到整个群体适应度求解完毕。最后根据计算结果确定是否更新最小路径信息。

4) 选择操作: 算法中选择机制采用适应度比例法(即赌轮选择)实现选择。设群体大小为 n, 其中个体 i 的适应度  $f_i$ , 则 i 被选择的概率  $P_i$  为:

$$P_i = f_i / \sum_{j=1}^M f_j \quad (1)$$

概率  $P_i$  反映了个体 i 的适应度在整个群体的个体适应度总和中所占的比例。按式(1)计算出群体中各个个体的选择概率。

5) 交叉操作: 文中采用一点交叉, 但因路径交叉以后出现实际不存在的路径, 所以对交叉采取一定的限制措施, 防止交叉后出现不存在的路径。并对生成的新个体做相应的修正, 使其成为满足条件的新个体。

6) 变异操作: 文中采用常规变异策略, 变异后再根据确定的位对原来的个体进行修正, 防止出现不存在的路径, 避免程序发生异常。

7) 程序进行选择与交叉过程中, 根据运行前指定的参数, 对群体中一定概率的最优个体进行了直接保留, 从而大大提高了优秀个体被保留的概率, 提高了算法的搜索效率。

#### 4.4.2 算法的分析与改进

利用遗传算法求解最短路径问题具有与传统方法不同的效益, 它可以获得最优解, 解决传统算法设计方法所不能解决的问题。该算法的复杂度为:  $O(n2^n)$ 。对于遗传算法的改进体现在基于惩罚函数修正方法和译码方法上, 通过修正得到最优解。

### 4 各种算法设计方法比较

通过以上对最短路径问题的求解分析, 我们可以看到各种算法设计方法有各自不同的特点, 针对不同的问题领域, 它们有不同的效率。对于求解最短路径问题, 各算法的时间复杂、优缺点以及改进方法的比较如表 1 所示。

#### 5 结束语

以上是对动态规划, Dijkstra 算法, A\* 算法, 遗传算法这四种常用算法设计方法原理的概要介绍, 并具体应用于求解最短路径问题中, 以及对各种设计方法的算法分析。通过论述研究, 得出各种算法在求解特定问题领域的优缺点和改进方法。

表 1 各种算法设计方法比较

设计方法	时间复杂度	优点	缺点	改进方法
动态规划	$O(\min(nc, 2^n))$	可以求得全局最优解	速度慢	建立动态规划递归方程
Dijkstra 算法	$O(n^2)$	求的全局最优解	节点多时速度很慢	采用堆排序等
A* 算法	比 Dijkstra 算法时间复杂度少	智能算法, 能求出接近最优解的解, 速度较快	算法不成功, 难求得最优路径	改进启发式函数
遗传算法	$O(n2^n)$	能够获得最优解, 解决传统算法不能解决的问题	速度较慢, 算法不成熟	基于惩罚函数修正方法和译码方法

(下转第 1412 页)

程/部件”菜单命令选择 Microsoft Script Control 1.0 选项, 添加 Script-Control 控件。实现步骤如下:

- 1) 将 ScriptControl 控件添加到工具箱;
- 2) 在窗体上放置 ScriptControl 控件;
- 3) 设置 ScriptControl 控件的 Language 属性为 vbscript; 利用 Eval 方法使用户可以根据实际需要, 在界面输入需要优化的方程。

3 应用实例

针对工程技术中的许多问题, 建立的目标函数常可归结为函数  $f(x)$  的形式, 其最优化设计的数学模型可简化为:  $\min f(x), x \in E^n$  或  $\max f(x), x \in E^n$ 。选取文献[1]上的一个例题进行验证。利用黄金分割法求目标函数为  $f(x) = x^2 - 7x + 10$  的最优解。已知初始区间为  $[2, 8]$ , 取迭代精度  $\epsilon = 0.35$ 。

在软件中选取黄金分割法优化界面, 输入已知条件, 如图 2 所示。观察输出结果, 可以得到极值点  $x = 3.58344$ , 极值  $f(x) = -2.24304$ 。根据软件优化结果, 发现和理论计算结果一致。

4 结束语

机械优化设计软件可以帮助工作设计人员简化计算, 缩短设计周期。但是, 每一种优化方法都具有一定的局限性, 在设计时应考虑求解问题的规模、数学模型的特性和复杂程度, 及各类优化方法的特点和应用范围, 选取合适的算法进行优化。

参考文献:

[1] 陈秀宁. 机械优化设计[M]. 杭州: 浙江大学出版社, 2002.  
[2] 孙靖民. 机械优化设计[M]. 北京: 机械工业出版社, 2004.  
[3] 孟兆明, 常德功. 机械最优设计技术[M]. 北京: 化学工业出版社, 2002.  
[4] 施光燕, 董加礼. 最优化方法[M]. 北京: 高等教育出版社, 2003.  
[5] 裴喜春, 王健, 周根宝. Visual Basic 程序设计教程[M]. 内蒙古: 内蒙古大学出版社, 2002.



图 2 黄金分割法优化界面

(上接第 1405 页)

参考文献:

[1] 余祥宜, 崔国华, 邹海明. 计算机算法基础[M]. 武汉: 华中科技大学出版社, 2003.  
[2] 王志和, 凌云. Dijkstra 最短路径算法的优化及其实现[J]. 软件时空, 2007(11).  
[3] 王凌. 智能优化算法及其应用[M]. 北京: 清华大学出版社, 2001.  
[4] 陈圣群, 滕忠坚, 洪亲, 等. 四种最短路径算法实例分析[J]. 开发研究与设计技术, 2007(7).  
[5] 梁娟, 郭军丽, 魏勇. 利用动态规划算法求解最短路径[J]. 河南机电高等专科学校学报, 2006(5).  
[6] 江务学, 李成银, 李黎明. 一种基于遗传算法的网络最短路径的求解[J]. 沙洋师范高等专科学校学报, 2007(3).

作者：[周先曙](#)  
作者单位：[中国人民解放军75742部队, 广东广州, 510515](#)  
刊名：[电脑知识与技术](#)  
英文刊名：[COMPUTER KNOWLEDGE AND TECHNOLOGY](#)  
年，卷(期)：2010, 6 (6)  
被引用次数：5次

## 参考文献(6条)

1. 余祥宣;崔国华;邹海明 [计算机算法基础](#) 2003
2. 王志和;凌云 [Dijkstm最短路径算法的优化及其实现](#) 2007 (11)
3. 王凌 [智能优化算法及其应用](#) 2001
4. 陈圣群;滕忠坚;洪亲 [四种最短路径算法实例分析](#) 2007 (07)
5. 梁娟;郭军丽;魏勇 [利用动态规划算法求解最短路径](#)[期刊论文]-[河南机电高等专科学校学报](#) 2006 (05)
6. 江务学;李成银;李黎明 [一种基于遗传算法的网络最短路径的求解](#)[期刊论文]-[沙洋师范高等专科学校学报](#) 2007 (03)

## 本文读者也读过(1条)

1. [柴登峰](#). [张登荣](#) [前N条最短路径问题的算法及应用](#)[期刊论文]-[浙江大学学报\(工学版\)](#) 2002, 36 (5)

## 引证文献(5条)

1. 田杰. [朱晓立](#). [李璐明](#) [城市交通诱导系统路径优化方法研究](#)[期刊论文]-[信息通信](#) 2013 (9)
2. 钟瑛. [陈凌峰](#). [朱顺痣](#) [改进A\\*算法在游戏地图路径搜索中的应用研究](#)[期刊论文]-[网络安全技术与应用](#) 2013 (8)
3. [严晓风](#). [陆济湘](#). [唐双平](#) [基于Floyd算法的校园最短路径问题分析与实现](#)[期刊论文]-[武汉理工大学学报 \(信息与](#)  
[管理工程版\)](#) 2012 (6)
4. [李冉冉](#) [基于蚁群算法的交通流诱导设计与实现](#)[期刊论文]-[计算机与现代化](#) 2011 (11)
5. [廖诗琪](#). [汪云龙](#) [单点交叉口交通信号配时模型及遗传算法](#)[期刊论文]-[公路与汽运](#) 2012 (3)

引用本文格式：[周先曙](#) [最短路径问题及其解法研究](#)[期刊论文]-[电脑知识与技术](#) 2010 (6)