

在嵌入式系统中求解无约束的现实 TSP 问题

胡军国^{1,2}, 孙毅¹, 祁亨年²

HU Jun-guo^{1,2}, SUN Yi¹, QI Heng-nian²

1. 浙江工业大学 机械制造及自动化教育部重点实验室, 杭州 310032

2. 浙江林学院 信息工程学院, 浙江 临安 311300

1. The MOE Key Laboratory of Mechanical Manufacture and Automation, Zhejiang University of Technology, Hangzhou 310032, China

2. College of Information Engineering, Zhejiang Forestry University, Lin'an, Zhejiang 311300, China

E-mail: hawkhjg@163.com

HU Jun-guo, SUN Yi, QI Heng-nian. Solving non-restraint realistic TSP problem in embedded system. *Computer Engineering and Applications*, 2008, 44(6): 225-227.

Abstract: The paper introduces to use simulated annealing algorithm and combine higher-probability -nearest-neighbor-algorithm-depth-first-search-algorithm to solve non-restraint TSP problem, and uses the improved depth-first-search-algorithm to create database structure of TSP subset. And the algorithms are successful used in embedded electronic guide system and bring to a good effect.

Key words: non-restraint realistic TSP problem; simulated annealing algorithm; higher-probability priority; depth first search; embedded electronic guide system

摘要: 介绍了利用模拟退火算法思想, 结合大概率最近邻深度搜索算法解决无约束的 TSP 问题, 以及用改进的深度优先遍历建立 TSP 子集数据结构和求解, 并成功地应用到嵌入式电子导游系统中, 取得良好效果。

关键词: 无约束实现 TSP 问题; 模拟退火算法; 大概率优先; 深度优先搜索; 嵌入式电子导游系统

文章编号: 1002-8331(2008)06-0225-03 **文献标识码:** A **中图分类号:** TP18

TSP 问题(Traveling Salesman Problem)是一类经典的组合优化问题。即给定 n 个城市以及各城市之间的距离, 要求找到一条遍历所有城市且每个城市只被访问一次的回路, 使得总路线距离最短。其数学描述表达如下:

设有 n 个城市 $C = \{C_1, C_2, C_3, \dots, C_n\}$, 其中任意两个城市的距离记为 $d(C_i, C_j)$, 求一条经过 C 中所有城市一次且仅一次的一条路径 $(C_{n(1)}, C_{n(2)}, C_{n(3)}, \dots, C_{n(N)})$, 使得闭合路径:

$$\sum_{i=1}^n d[C_{n(i)}, C_{n(i+1)} \bmod n] \text{ 为最小}^{[1,2]}$$

上文提到的 TSP 问题与本文将要阐述的无约束的现实 TSP 问题不同, 为了区分, 本文称其为传统 TSP 问题。传统 TSP 问题是 NP 难问题^[1], 现在还不存在有效的最优解, 但也有很多可行的近优解, 如: 禁忌搜索算法、蚁群算法、遗传算法、神经网络算法、模拟退火算法等^[3-5]。

在传统的 TSP 问题求解中, 要求每个城市都需经过, 且仅经过一次的哈密尔顿回路, 对图有比较高的要求。而在现实生活中, 城市之间并不一定能构成哈密尔顿回路, 并且, 旅行者为了行程最近, 也可以回走; 甚至, 游客不是游遍所有的 n 个城市, 而是行走其中的 $m(m < n)$ 个城市。所以传统的 TSP 问题求解法在现实生活中并不能很好适用。如图 1 所示, 在某旅游景

区有 5 个景点, ABCDE, 按传统 TSP 访问 ABDE 就不能返回到 A, 在实现中可按 ABDEDBACA 来访问所有景点。有时游客只对 ADE 感兴趣, 可按 ABDEDBA 来访问, 以传统 TSP 根本无法求解。

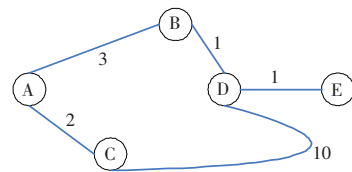


图 1 非传统 TSP 问题

所以本文提出的无约束的现实 TSP 问题, 与传统 TSP 问题不同:

- (1) 每个城市都允许回走。
- (2) 行走的回路允许不是哈密尔顿回, 而是任何回路。
- (3) 行走的城市可能是所有城市的子集。

1 无约束的 TSP 问题求解

对于传统的 TSP 问题解法常用的有蚁群算法、遗传算法、神经网络算法等智能算法, 这些算法计算量比较大, 在运算过程中还会出现小数。而大部分嵌入式系统的芯片都不带浮点运

基金项目: 2007 年度浙江省科技支撑和引导计划面上重点项目 (No. 2007C21004)。

作者简介: 胡军国 (1978-), 男, 讲师, 主要研究方向是计算机嵌入系统、嵌入式 GIS。

收稿日期: 2007-06-20 修回日期: 2007-08-30

算,CPU 运行速度相对较慢,所以在嵌入式系统中运用这类算法不可取。经过大量的对比和实践,选用模拟退火思想,结合大概率最近邻深度搜索算法求解。

1.1 无约束的 TSP 数学模型建立

设 $G=(V,E), V=\{v_1,v_2,\dots,v_m\}, E=\{e_1,e_2,\dots,e_n\}$ ^[6], 寻找一条从 v_1 出发的回路 $P=v_1e_{i_1}v_2e_{i_2}v_3\dots e_{i_k}v_k\dots e_{i_l}v_l\dots v_1$, 使得 P 最短,且经过 V 中的所有点。同时,允许 e_{i_k},e_{i_l} 相等;也允许 v_{i_k},v_{i_l} 相等。

1.2 无约束的 TSP 算法实现

在嵌入式系统中,为了降低算法计算量,避免出现浮点运算,利用模拟退火算法(simulated annealing algorithm),通过内外两个循环,内循环保证局部最优,外循环保证全局最优。外循环持续进行“产生新解-判断-接受/舍弃”的迭代过程,使解逐步接近最优;内循环利用改进的最近邻算法(Nearest-neighbor algorithm),最近邻算法见参考文献[7,8]。改进的最近邻算法就是大概率最近邻深度搜索算法(higher probability nearest-neighbor algorithm depth first search algorithm),最邻近点优先,但也不放过其它可能的点,从而防止模拟退火算法在迭代过程中遗漏最优解。

1.2.1 模拟退火算法求解

模拟退火算法(SA)见参考文献[3-5],最早在 1953 年由 Metropolis 提出,到 1983 年由 Kirkpatrick 等人成功地应用到组合最优化问题中。算法的基本思想来源于固体退火降温过程,当固体加热时,分子呈现出自由运行状态,随着温度下降分子逐步趋于有序,最后以一定的结构排列。

本算法搜索策略上用的是最近邻算法就是大概率最近邻深度搜索算法(HPNNADFS),在温度下降中连续迭代,舍去非优解,接收最优解,使目标逐步接近最优。具体算法如下:

- (1) 给定初始温度 t_0 和初始函数值 $hpnnadfs(t_0)$ 。
- (2) 得到新的温度 $t_{k+1}=tk+\Delta t$, (t_k 为第 k 次迭代时的温度), 计算 $\Delta dist=hpnnadfs(t_{k+1})-hpnnadfs(t_k)$ 。
- (3) 若 $\Delta dist < 0$ 接收新的函数值 $hpnnadfs(t_{k+1})$, 记录新温度 t_{k+1} , 否则舍弃。
- (4) 温度到达预先设定的值或函数值达到满意结果进入第(5)步, 否则继续第(2)步。
- (5) 获得最终函数值 $hpnnadfs(t_{k+1})$, 和最优温度值 t_{k+1} , 算法结束。

算法在温度下降中,是通过循环变量 i 递进来控制温度下降,在每个温度中,通过 $hpnnadfs$ 函数获得新的路径值。下面介绍 $hpnnadfs$ 函数实现原理。

1.2.2 大概率最近邻深度搜索算法(HPNNADFS)

最近邻算法是一种贪心算法,它能保证局部最优,但随着算法的进行,会导致离返回点越走越远的后果,不能保证全局最优。通过对实际路径分析,发现大部分情况下一步应该选择最近的节点进行搜索,只有小概率情况往其它节点搜索。对图 1 建立邻接表数据结构^[8],如图 2。

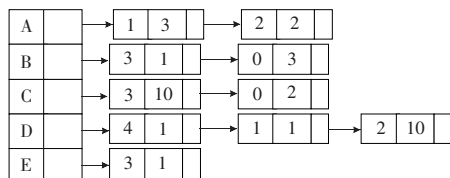


图2 邻接表数据结构

设节点 i 的度为 d_i ,以最近邻搜索的概率为 $1/2+1/(d_i-1)$,

以 $1/2-1/(d_i-1)$ 的概率随机搜索其它节点,(其中 $d_i>1$)。从 A 出发,以 0.75 的概率访问 C,0.25 的概率访问 B,假设访问 C(当然也有可能访问 B),接着访问 D 和 E,当访问到 E 时,一次大概率最近邻深度搜索结束,接着访问下一个没访问过的节点,通过回溯法^[8],确定下一个要访问的节点是 B。然后从 E 出发,利用改进的 Dijkstra 算法^[3,8],确定 E 到 B 的最短路径为 EDB,当访问 B 后,所有节点都已访问,再一次利用改进的 Dijkstra 算法确定 B 返回到 A 的最短路径为 BA。通过一次 HPNNADFS 算法,找到一条从 A 出发的回路:ACDEBA。为了提高 HPNNADFS 算法的效果,从每个节点出发,进行一次 HPNNADFS 算法,把所获得的最短路径作为这轮 $hpnnadfs$ 函数值。

2 TSP 子集问题求解

2.1 TSP 子集问题提出

图 1 是某旅游景区的 5 个景点,某游客只对 ADE 感兴趣,他从 A 出发,访问 DE 后回到 A 点,求一条最短的回路。

2.2 数据模型建立

设 $G=(V,E), V=\{v_1,v_2,\dots,v_m\}, E=\{e_1,e_2,\dots,e_n\}; G'=(V',E'), V'=\{v'_1,v'_2,\dots,v'_m\}, E'=\{e'|e'=d(v'_i,v'_j)\}$, 其中 $v'_i \in V', v'_j \in V', d(v'_i,v'_j)$ 为 v'_i 到 v'_j 的 DFS 最短路径。求访问 G' 中所有节点并回到起始点的最短路径。这里的 DFS 最短路径是指从 v'_i 出发,进行深度优先遍历,遇到任一点 $v'_j \in V'$ 即停止,选择最近的 v'_i 到 v'_j 的路径。

2.3 算法实现

从 V' 中的点 v_i 出发,进行深度优先遍历(DFS)^[8],直到遇到 $v_j(v_j \in V')$ 停止,所得的距离为 d_{new} , 路径为 p_{new} , 判断 d_{new} 是否小于原来的 $d(v_i,v_j)$, 若是,则 $d(v_i,v_j)=d_{new}$, 路径变为 p_{new} 。否则 $d(v_i,v_j)$ 不变。对所有 $v \in V'$ 的点进行递归 DFS, 得 TSP 子集: ADE, 其中 AD 距离为 4, DE 为 1。TSP 子集数据结构见图 3。

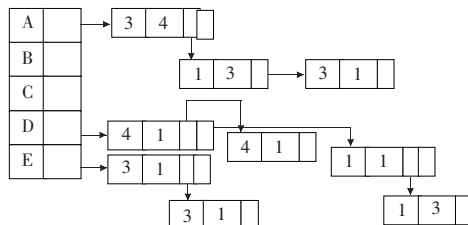


图3 TSP 子集数据结构

3 在现实旅游景区例证

本文所介绍的算法思想主要是用在手持式电子导游系统中,系统所用的芯片是 S3C2410,操作系统是 Linux-2.6.14,景区为杭州西湖景区,见图 4。



图4 杭州西湖景区

由于在实际景区中, 一些路会交叉, 交叉口会影响路径走向, 所以图 4 不能直接使用, 要对交叉路口进行改造, 使它成为虚拟景区, 如图 5 所示。

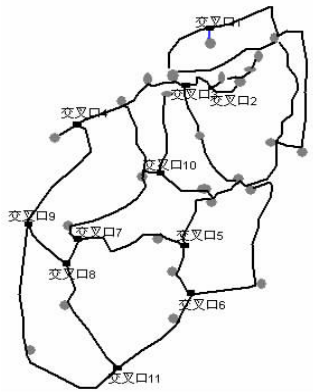


图 5 交叉路口改造为虚拟景区

对西湖景区景点改造后, 有 26 个景点和 11 个虚拟景点, 共 37 个点。通过模拟退火降温 2 500 次, 共进行 3 次运算, 得到访问 37 个点所形成的回路分别为: 43 050 m、42 840 m、42 620 m, 所花时间都为: 70 min 左右。从结果看基本上得到比较稳定的值, 取 42 620 m 的行程作为最终路径, 保存到配置文件中, 以后要访问这 26 个景点, 只要读取配置文件路径即可, 无需重新计算。

但在 TSP 子集问题中, 游客访问的是 26 个景点的子集, 随游客的不同而动态变化, 对此就不能先长时间运算, 保存结果, 只能实时运算。在实例中, 随机选取 13 个景点, 降温 10 000 次, 进行三次运算得: 26 650 m、25 410 m、25 410 m, 所花时间

约 15 s。所得结果比较稳定, 时间也能承受。

4 结论

本文采用模拟退火思想, 结合大概率最近邻深度搜索策略, 解决无约束 TSP 问题; 用改进的深度优先遍历算法, 建立 TSP 子集数据结构, 实现遍历全集中的子集问题。并把这两种方法用到实现的杭州西湖景区中, 在手持式电子导游系统中发挥很好作用, 取得良好效果。

参考文献:

- [1] 蔡之华, 彭锦国, 高伟, 等. 一种改进的求解 TSP 问题的演化算法[J]. 计算机学报, 2005, 28(5): 823-828.
- [2] 张贵军, 吴惕华, 叶蓉. CTSP 问题稳态小生境算法的研究及仿真实现[J]. 系统仿真学报, 2004, 8: 1692-1696.
- [3] 邢文训, 谢金星. 现代优化计算方法[M]. 2 版. 北京: 清华大学出版社, 2005.
- [4] 张光澄. 非线性最优化计算方法[M]. 北京: 高等教育出版社, 2005.
- [5] Skiscim C C, Golden B L. Optimization by simulated annealing: a preliminary computational study for the TSP[C]//Proceedings of the 15th Conference on Winter Simulation, Dec 1983: 523-535.
- [6] 卢开澄, 卢华明. 图论及其应用[M]. 2 版. 北京: 清华大学出版社, 1995.
- [7] McNames J.A fast nearest-neighbor algorithm based on a principal axis search tree[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001, 9(23): 964-976.
- [8] 严蔚敏, 吴伟民. 数据结构(C 语言版)[M]. 北京: 清华大学出版社, 2002.

(上接 201 页)

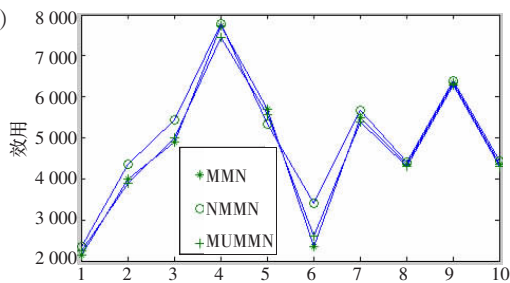


图 4 10 次仿真实验中企业 1 的利润对比

图 4 给出了企业 1 在文献[6]给出的 MMN、文献[7]给出的基于边际效用的市场机制协商方法 MUMMN 以及本文方法 NMMN 下的对比情况。由图 4 看出, MMN 和 MUMMN 性能接近, 但 MUMMN 具有更好的私密性和更大的协商代价。NMMN 性能略高于 MMN 和 MUMMN。这是因为企业智能体利用了神经网络预测原料价格, 对对手和市场环境表现出一定的预动性和前瞻性, 使得协商能够较快的收敛, 提高了资源配置的有效性。图 5 给出了三者的协商代价对比, 由于 NMMN 具有预动性, 能够较快的收敛, 因此相应的降低了协商回合数和通信代价。

参考文献:

- [1] 韩伟, 陈优广. 电子市场买方结盟的利益分配及其结盟策略[J]. 计算机集成制造, 2007, 13(12): 2487-2492.
- [2] Spier E. From reactive behavior to adaptive behavior: motivational

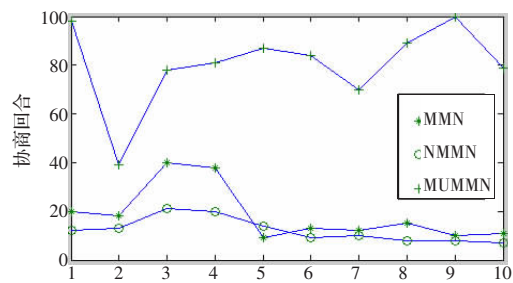


图 5 10 次仿真企业 1 的协商代价对比

models for behavior in animals and robots[D]. Oxford University, UK, 1997.

- [3] Davis S R. Negotiation as a metaphor for distributed problem solving[J]. Artificial Intelligence, 1983, 20: 63-109.
- [4] Fatima S, Wooldridge M, Jennings N R. Optimal agendas for multi-issue negotiation[C]//Proc 2nd Int Joint Conf on Autonomous Agents and Multi-Agent Systems, Utrecht, Netherlands, 2005: 635-642.
- [5] 韩伟. 电子市场环境下的多智能体学习与协商[D]. 上海: 华东师范大学, 2006-04.
- [6] Han Wei. Metropolitan pollution reduction by intelligent agent[J]. Wuhan University Journal of Natural Sciences, 2004(9).
- [7] 韩伟. 基于多智能体协商的电子市场原料配置模型[J]. 计算机应用, 2006(12).
- [8] 韩伟. 一种强调私密性的多智能体协商模型[J]. 计算机应用, 2007(4).