

文章编号:1007-6735(2010)05-0466-05

求解旅行商问题的混合量子算法

傅家旗, 叶春明

(上海理工大学 管理学院, 上海 200093)

摘要: 提出了求解旅行商问题的混合量子算法(HQA). HQA 以量子计算为基础, 设计了移位解码, 解决了构造路径难的问题. 并采用微粒群算法的进化模式和跟踪保优模式, 构造了动态惯性权重使量子角更新、更有效, 增加了局部优化进行精细搜索. 对多个算例的测试结果表明, HQA 具备了求解旅行商问题的能力.

关键词: 混合量子算法; 旅行商问题; 优化

中图分类号: TP 18; TP 301.4

文献标志码: A

Hybrid quantum algorithm for solving traveling salesman problems

FU Jia-qi, YE Chun-ming

(Business School, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: Hybrid quantum algorithm (HQA) was put forward to solve traveling salesman problems. On the basis of quantum computing, displacement decoding was designed which enables the algorithm to overcome the difficulties in route configuration. Evolutionary mode and best tracing and keeping mode, derived from particle swarm optimization algorithm, were adopted, dynamic inertia weight was configured to update quantum angles effectively, and local optimization was taken to carry out lean search. The capability of HQA to solve traveling salesman problems was demonstrated by the results of several examples.

Key words: hybrid quantum algorithm; traveling salesman problems; optimization

旅行商问题(TSP)描述的是一位旅行商想访问各个城市一次并返回出发城市, 选择何种路线能使总行程最短的问题. TSP 具有悠久的历史, 最早的记录可追溯到 1759 年欧拉研究的骑士周游问题^[1]. 如今, TSP 在许多工程中都有体现, 小到芯片制造系统中决定连接半导体芯片和印刷线路板板间的金线行走路径, 大到楼宇中的管道铺设, 甚至奥运会火炬传递行程都可归结为 TSP, 这些 TSP 如能处理得当, 必给整个工程带来裨益. 因此, 能够解决好 TSP 也就相当于

掌握了解决此类问题的钥匙. 鉴于 TSP 的优化算法其复杂度随城市数量的增加呈指数增长, 一旦城市数目庞大, 即使性能最好的计算机也要花相当大的时间代价. 从计算复杂性来说, TSP 是 NP 完全(non-deterministic polynomial complete)的, 也就是说无法确定求解 TSP 的多项式时间算法.

量子计算是研究量子现象实现计算的一个构想, 旨在构造精良的量子算法以及研制可执行这些算法的量子计算机. 在 20 世纪 90 年代, 已经有将进

收稿日期: 2008-12-08

修返日期: 2009-04-30

基金项目: 高等学校博士点基金资助项目(20093120110008); 上海市重点学科资助项目(S30504)

作者简介: 傅家旗(1984-), 男, 硕士研究生. E-mail: iejqfu@126.com

化计算与量子计算相结合的尝试,其成果之一就是利用量子特性在普通计算机上实现量子进化计算. 具有概率搜索能力的进化算法虽仅有概率意义上的全局收敛性,由于在可接受的时间范围内仍有全局收敛的可能,故能应用于 TSP 等组合优化问题. 目前,量子优化算法应用于组合优化问题的可行性已得到证实,但计算的效率和效果主要是由算法设计决定,仍有进一步研究的价值. 文献[2]设计了灵活的编码方式,既通过量子特性产生可行路径,又可将解码后的信息应用于旋转门操作. 文献[3]运用较为直观的矩阵编码方式,有利于适应值计算. 但这两种编码方式都需要在运算中进行修正操作,且编码需要的存储量较大. 本文的混合量子算法(HQA)与之不同,其特点在于将量子比特码解码为路径的思想能节省存储空间;借助微粒群算法(PSO)更新量子角以推动种群进化;设计的量子比特映射交叉(QMX)及构造的优化规则有助于增强算法的鲁棒性.

1 混合量子算法

1.1 编码和解码

在量子计算理论中,量子比特不仅能处于 0 或 1 的状态,还能处于 0 和 1 的叠加态,而终态取决于观察的结果. 一个量子比特码随着观察的不同,会塌缩为不同的状态,因而具有并行计算的潜力. 量子比特编码是基于量子比特的相干性和叠加性设计的,假设第 t 代的种群为 $Q(t) = \{q_1^t, q_2^t, \dots, q_n^t\}$, q_i^t 即为染色体,具体形式可以描述为

$$q_i^t = \begin{bmatrix} \alpha_{i1}^t & \alpha_{i2}^t & \dots & \alpha_{id}^t \\ \beta_{i1}^t & \beta_{i2}^t & \dots & \beta_{id}^t \end{bmatrix} \quad i = 1, 2, \dots, n \quad (1)$$

式中, n 为种群大小; D 为量子染色体的长度; $(\alpha_{id}^t, \beta_{id}^t)^T$ 表示一个量子比特, $d = 1, 2, \dots, D$; $|\alpha_{id}^t|^2$ 和 $|\beta_{id}^t|^2$ 分别是塌缩为 0 态和 1 态的概率,满足 $|\alpha_{id}^t|^2 + |\beta_{id}^t|^2 = 1$.

随着 $|\alpha_{id}^t|^2$ 或 $|\beta_{id}^t|^2$ 逐渐趋于 1 或 0,量子染色体收敛于一个状态,这时种群的多样性也逐渐消失^[4]. 对 q_i^t 进行一次观察,可得到 0-1 码 $P(t)$,具体操作是产生一个 $[0, 1]$ 的随机数 r ,若 $r \geq |\alpha_{id}^t|^2$,相应量子比特位出 1;否则,出 0.

由于生成的个体为一条路径,须对已得的解码 $P(t)$ 再行解码,将这一步解码称为移位解码. 假设当前有一条可行路径(即个体),定义 0 取当前个体的首基因,1 取当前个体的第 2 位基因,按此思路,从父代个体中一一取出基因组成子代个体,取出的基因即从

父代中剔除,以进行下一位的选择,如表 1 所示.

表 1 移位解码
Tab.1 Displacement decoding

名称	码								
父代个体	4	5	1	3	8	6	9	7	2
$P(t)$	0	1	1	0	1	1	1	0	0
子代个体	4	1	3	5	6	9	7	8	2

1.2 量子进化

在混合算法中,量子进化是推动种群进化的主要手段并主宰搜索行径的大方向,而在量子进化中起决定性作用的就是量子角,因而算法的优劣在很大程度上是由量子角的进化机制所决定的. 一般量子优化算法中,量子角的更新是通过考察多个变量确定量子旋转门,并经量子旋转门对 $Q(t)$ 进行矩阵运算来实现量子比特的更新. 这里对于个体的进化,采用形如 PSO 算法的变化方式,既保证量子角的更新有较为科学的规律可循,又无需对量子角进行区块分割,使变化更加灵活,实现智能化,表达式为

$$v_{id}^{t+1} = \omega v_{id}^t + c_1 r_{1d} (p_{id}^t - \theta_{id}^t) + c_2 r_{2d} (p_{gd}^t - \theta_{id}^t) \quad (2)$$

$$\theta_{id}^{t+1} = \theta_{id}^t + v_{id}^{t+1} \quad (3)$$

$$\alpha_{id}^{t+1} = \cos(\theta_{id}^{t+1}) \quad (4)$$

式中,下标 d 为个体的维度;下标 i 表示此为第 i 个个体; θ_{id}^t 为量子角; v_{id}^t 为量子角增量; p_{id}^t (或 pbest) 为个体独自进化过程中出现的最优个体; p_{gd}^t (或 gbest) 为种群进化过程中出现的最优个体; ω 为惯性权重; c_1, c_2 为加速常数; r_{1d}^t, r_{2d}^t 为 $[0, 1]$ 的随机数.

1.3 动态惯性权重 ω

量子角的改变对量子优化的影响巨大,故需对式(2)中影响增量 v_{id}^{t+1} 取值的各个组分的参量进行讨论. 相关文献给出了第二、三部分参量的参考取值范围,而第一部分 ω 的取值最为关键,不仅决定了上一代量子角的增幅在多大程度上被保留,还间接决定了在 v_{id}^{t+1} 中第一部分所占的比重,确定其值的方法设计如下:

a. 计算距离. δ_{fi}^t 和 δ_{Gi}^t 表示个体 i 的适应值与种群平均适应值 f_{avg}^t 及种群最优适应值 G^t (或 gbest) 的距离

$$\delta_{fi}^t = (f_i^t - f_{avg}^t) / f_{avg}^t \quad (5)$$

$$\delta_{Gi}^t = (f_i^t - G^t) / G^t \quad (6)$$

b. 标准化. $\delta_{\min, fi}^t$ 和 $\delta_{\max, fi}^t$, $\delta_{\min, Gi}^t$ 和 $\delta_{\max, Gi}^t$ 分别表示 δ_{fi}^t 和 δ_{Gi}^t 中的最小值和最大值,根据式(7)和式(8)将求得的 δ_{fi}^t 和 δ_{Gi}^t 标准化为 $\delta_{sta, fi}^t$ 和 $\delta_{sta, Gi}^t$.

$$\delta_{sta, fi}^t = (\delta_{fi}^t - \delta_{\min, fi}^t) / (\delta_{\max, fi}^t - \delta_{\min, fi}^t) \quad (7)$$

$$\delta_{sta, Gi}^t = (\delta_{Gi}^t - \delta_{min, Gi}^t) / (\delta_{max, Gi}^t - \delta_{min, Gi}^t) \quad (8)$$

c. 确定惯性权重. $d\omega_i^t$ 作为最终的惯性权重

$$d\omega_i^t = \delta_{sta, \bar{f}_i}^t + \delta_{sta, Gi}^t \quad (9)$$

d. 修正. 确定 $d\omega_i^t$ 以后, 按

$$d\omega_i^t = \begin{cases} -\min \omega f & d\omega_i^t \leq \min \omega f \\ d\omega_i^t & \min \omega f < d\omega_i^t < \max \omega f \\ \max \omega f & d\omega_i^t \geq \max \omega f \end{cases} \quad (10)$$

对 $d\omega_i^t$ 进行修正. 修正的作用是为了防止算法搜索过分发散, 使种群中任何个体的速度增幅在控制范围内, 保证种群向最优极值点逼近.

2 混合量子算法中的其他优化机制

2.1 初始化

优化算法的优劣在一定程度上取决于初始解, 如果初始种群的邻域恰好与最优个体的邻域存在交集, 搜索到最优个体的概率将大大增加. 假设所有操作都以城市 1 为起点, 设计了一种启发式的初始化方法, 计算步骤如下:

a. 从权重矩阵中找到最小权, 由于存在对称, 以第一次找到的为准, 记该权的行号为 r , 以 r 作为起点, 记权的列号为 c , 以 c 作为下一点, 将该权划去, 并根据对称性将另一个权也划去;

b. 搜索第 c 列中的最小权(被划去的权不考虑), 同样以第一次找到的为准, 记录该权的行号为 r , 如果该权的 r 在所构造的路径中已出现过, 将该权及对称权划去, 重复步骤 b; 否则, 以 r 作为路径的下一点, 并将该权及对称权划去, c 即取 r ;

c. 重复步骤 b, 直到生成出完整路径;

d. 对完整路径进行调整, 使城市 1 位于起点.

根据上述步骤, 以 10 城市为例, 其权重矩阵列于表 2. N 为城市序号. 经计算可得完整路径为 2—10—5—4—6—3—1—9—8—7, 经过调整, 以城市 1 为起点的路径就成为 1—9—8—7—2—10—5—4—6—3—1. 由于城市 1 固定, 则不参与计算, 将 9—8—7—2—10—5—4—6—3 作为个体, 进行下面的初始化操作;

e. 以当前的完整路径作为基准, 以首基因“9”作为确定基因, 其余基因作为不确定基因. 从第 2 个基因“8”开始, 依次将不确定基因插入确定基因间的位置, 从原路径和新路径中找到最短的一条路, 并将当前的第一位不确定基因作为确定的, 重复步骤 e, 直到路径中的基因都确定下来.

f. 将步骤 e 中适应值排在前 n 位的个体作为

初始种群.

表 2 10 城市的权重矩阵

Tab.2 Weight matrix of 10 cities

N	N									
	1	2	3	4	5	6	7	8	9	10
1	∞	50	2	95	86	14	78	95	20	12
2	50	∞	44	51	67	57	74	51	52	1
3	2	44	∞	95	42	12	24	74	62	54
4	95	51	95	∞	5	19	86	74	92	67
5	86	67	42	5	∞	59	5	100	17	5
6	14	57	12	19	59	∞	80	99	34	53
7	78	74	24	86	5	80	∞	32	89	39
8	95	51	74	74	100	99	32	∞	11	75
9	20	52	62	92	17	34	89	11	∞	57
10	12	1	54	67	5	53	39	75	57	∞

2.2 局部优化

局部优化是细致化的搜索过程, 能较好地反应算法的优劣, HQA 主要采用两种局部优化方法.

局部优化 1 BELLMAN 提出的最优化理论^[5]指出, 最短路就是指路径中的任意两个城市间的距离都是最短的, 即解空间中仍存在能被优化的路径必非最短路. 由于两城市间的距离是确定的, 从最大距离入手, 对路径进行节约, 计算方法如下:

寻找当前路径中间距最长的两城市 t_1 和 t_2 , 如图 1 所示, 从权重矩阵中寻找与 t_1 距离较近的城市集, 对 t_2 也如法炮制. 考察两组离 t_1 和 t_2 较近的城市集是否存在交集, 仅对存在交集的情况操作.

当交集中存在的城市数为 1, 设该城市为 T , 且 T 与 t_3, t_4 相邻, 如图 1 所示. 分别将 t_1 和 t_2, T 和 t_3, T 和 t_4 断开, 使 t_1, t_2 与 T 相邻, t_3 和 t_4 相邻, 便得到新路径.

当交集中存在的城市数超过 1, 仿照上法, 逐一计算, 仅保留最短的那条路.

局部优化 2 局部优化 1 是根据 TSP 的行程特征展开搜索, 无法直接缩短路径, 却能在路径上接近极值点的值, 这就要求局部优化尽可能覆盖解空间. 因而运用局部优化 2 配合局部优化 1 的作用, 其优化方法和初始化过程中确定基因的方法相似, 展开邻域如表 3 所示.

2.3 量子映射交叉和隔离小生境多交叉

在混合量子算法中曾采用固定交叉位置的部分映射交叉(PMX)交叉算子配合微粒群算法的更新式促使个体进化^[6], PMX 具有保留最佳模式的特点, 但携带最优信息的基因会因位置选取的不恰当

而无法为种群注入新的活力,这里采用一种变化性更大的交叉算子——量子映射交叉(QMX)算子.交叉方法如下:根据 $P(t)$ 的取值,将 $P(t)$ 为0的基因作为模式保留,而对 $P(t)$ 为1的基因进行映射交叉.在表4中, $X(t)$ 和 $GX(t)$ 为原个体, $XF(t)$ 和 $XS(t)$ 为映射后的新个体.以 $XF(t)$ 的计算为例说明QMX的操作方法,首先将对应 $P(t)$ 为0的 $GX(t)$ 中的基因赋给 $XF(t)$ 对应位置上的基因,然后将 $X(t)$ 中不处在交换位置上的基因按顺序填入空缺处,起冲突的基因用映射方法消解. $XF(t)$ 第一个空缺应填入1,但1发生了冲突,1的映射为10,且不发生冲突,因而填入10,下一个空缺处填入6不发生冲突,剩余空缺的基因依照此法顺序推算,即可得到表4中的 $XF(t)$ 和 $XS(t)$.

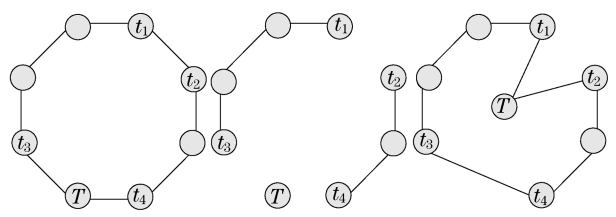


图1 局部优化1
Fig.1 Local optimization 1

表3 邻域结构
Tab.3 Neighborhood structure

名称	码		
初始个体	1	2	3
	2	1	3
展开邻域	3	1	2
	1	3	2

表4 QMX
Tab.4 QMX

名称	码									
$X(t)$	1	4	7	8	6	3	5	10	2	9
$P(t)$	1	0	0	0	1	1	0	0	0	1
$GX(t)$	2	4	3	5	8	6	9	1	7	10
$XF(t)$	10	4	3	5	6	2	9	1	7	8
$XS(t)$	3	4	7	8	9	6	5	10	2	1

出于变化性的考虑,按隔离小生境将种群等分,以子种群为单位,选择各种交叉算子,各自独立交叉.仿真时将种群分为4个子种群,分别采用QMX、单位置次序交叉(C1)、Non-ABEL和循环交叉(CX)这4种算子交叉.另外,采用逆序变异算子伴随交叉的进行.

2.4 模拟退火保留

在优化计算中,新老个体更替频繁,如果一味地将适应值好的个体给保留下来,容易使整个种群逐步向某个局部极值点逼近,最终难以跳出局优.而如果仅随迭代而舍弃旧个体,不顾适应值的优劣,亦难保证种群的质量.因此,保留策略既要使种群具有较强的突跳能力,还要力求种群得到稳步优化.模拟退火算法^[7]的保留机制按适应值的优劣一分为二考虑,优解一定被保留,而对于劣解按概率接受,具备了保证种群质量和种群突跳能力强的素质.

2.5 混合量子算法流程

混合量子算法流程如图2所示.

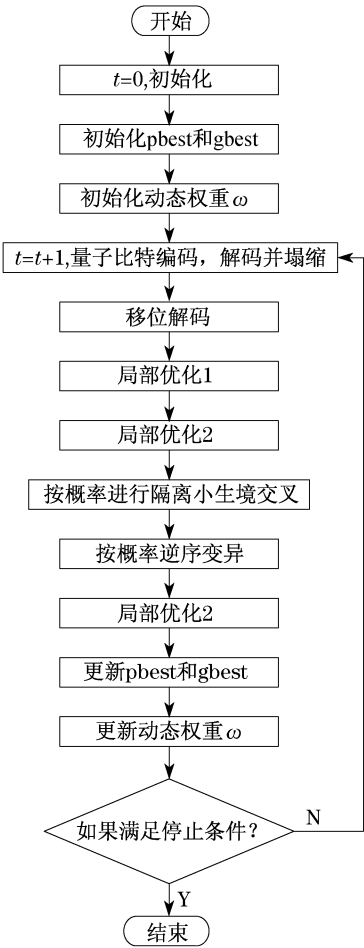


图2 HQA 算法流程图
Fig.2 Flowchart of HQA

3 仿 真

TSPLIB^[8]是国际通用的算例库,从中选取7个

不同数据规模的问题进行测试. 在 2.53 GHz Intel Core™2 Duo CPU, 2 GB RAM 的计算机上, 用 Visual C++ 6.0 开发了 HQA, 分别对每个问题进行了 20 次测试. 部分参数设置如表 5 所示, 测试的统计结果如表 6 所示. C_N 为城市数, F_B 为全局最优适应值, I 为预设进化代数, I_A 找到最短路的平均进化代数, C_0 为达到最优解的次数, F_A 为种群平均适应值, F_W 为种群中的最差适应值.

表 5 部分参数设置
Tab.5 Settings of parameters

参 数	取 值
种群规模 n	24
隔离小生境交叉概率 p_c	0.75
逆序变异概率 p_m	0.1
退火初温 t_s	进化代数
加速常数 c_1	2.0
加速常数 c_2	1.0

表 6 算例测试的统计结果
Tab.6 Statistical results of examples

TSP 算例	C_N	F_B	I /次	达到最优解的情况			
				I_A /次	C_0 /次	F_A	F_W
gr17	17	2 085	500	1.00	20	2 085.00	2 085
gr21	21	2 707	500	72.70	20	2 707.00	2 707
gr24	24	1 272	500	99.05	20	1 272.00	1 272
fri26	26	937	500	1.30	20	937.00	937
gr48	48	5 046	1 500	673.83	6	5 068.10	5 115
hk48	48	11 461	1 500	350.83	6	11 500.75	11 641
berlin52	52	7 542	1 500	442.00	14	7 584.05	7 745

在统计结果中, 前 4 个小规模算例, HQA 能够既快又好地求解. 随着城市数的增加, 从平均进化代数可知, 构造最优路径的计算代价也越来越大. 图 3 为 HQA 求解算例 gr48 得到最优解 11461 的迭代情况, 最优个体随着迭代的进行, 逐步向全局最优点靠拢, 直至找到最优点. F 表示适应值.

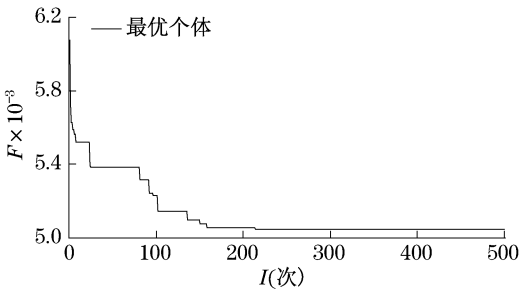


图 3 求解 gr48 的迭代情况
Fig.3 Iteration of optimizing gr48

4 结束语

TSP 的用途已远远超越了行程的意义, 许多工程中的技术问题也可抽象为 TSP. 高效而精确地求解 TSP 不仅是理论研究的需要, 更是工程中的各项资源达到合理配置的基本保障. 标准的优化算法在 TSP 问题的应用极为广泛, 混合优化策略是将来研究 TSP 的一个发展方向. 从量子计算出发, 结合多种优化技术的混合量子算法是一个初步尝试. 虽然

量子技术目前仍存在很多技术难关, 相信在不久的将来, 这些问题得到解决以后, 将会大大推动优化技术和优化算法的发展.

参考文献:

[1] [美]MICHALEWICZ Z, FOGEL D B. 如何求解问题——现代启发式方法[M]. 曹宏庆, 译. 北京: 中国水利水电出版社, 2003: 143 – 170

[2] 王宇平, 李英华. 求解 TSP 的量子遗传算法[J]. 计算机学报, 2007, 30(5): 748 – 755.

[3] 杨丽, 李平, 秦亚玲. 改进的量子进化算法及其在 TSP 问题中的应用[J]. 信息与电子工程, 2006, 4(6): 412 – 416.

[4] HAN K H, KIM J H. Genetic quantum algorithm and its application to combinatorial optimization problem [C]// IEEE Proceedings of the 2000 Congress on Evolutionary Computation. Piscataway: IEEE Press, 2000: 1 354 – 1 360.

[5] 朱自强, 王龙德. 运筹学基础教程[M]. 四川: 成都科技大学出版社, 1997: 215 – 228.

[6] 傅家旗, 叶春明, 谢金华. 混合量子算法及其在 flow shop 问题中的应用[J]. 计算机工程与应用, 2008, 44(20): 48 – 50.

[7] 邢文训, 谢金星. 现代优化计算方法[M]. 第 2 版. 北京: 清华大学出版社. 2005: 77 – 81.

[8] Office Research Group Discrete Optimization. TSPLIB [DB/OL]. <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>, 2008 – 08 – 06/2008 – 11 – 15