

文章编号: 1001-0920(2011)12-1855-06

基于图知识迁移的蚁群算法参数选择

王雪松, 潘 杰, 程玉虎

(中国矿业大学 信息与电气工程学院, 江苏 徐州 221116)

摘 要: 同时考虑蚁群算法的所有运行参数, 提出一种基于图知识迁移的蚁群算法参数选择方法. 首先, 将包含知识(蚁群算法的运行参数)的源任务映射到一个高维的迁移空间, 并通过迁移权值连接不同的源任务, 构造一个模型迁移图; 然后, 扩展模型迁移图使其包含目标任务, 并利用图论的知识学习迁移函数; 最后, 通过最小二乘法自主地给目标任务分配一个优化的运行参数组合. 机器人路径规划问题的仿真结果验证了该方法的智能性、快速性与合理性.

关键词: 蚁群算法; 参数选择; 图知识迁移; 路径规划

中图分类号: TP18

文献标识码: A

Parameters selection for ant colony algorithms based on graph knowledge transfer

WANG Xue-song, PAN Jie, CHENG Yu-hu

(School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou 221116, China. Correspondent: WANG Xue-song, E-mail: wangxuesongcumt@163.com)

Abstract: A kind of parameters selection for ant colony algorithms (ACAs) based on graph knowledge transfer is proposed, where all of running parameters are taken into account simultaneously. Firstly, all source tasks containing knowledge (running parameters for ACAs) are mapped onto a high-dimensional transfer space, and transfer weights are used to connect these source tasks. In this way, a model transfer graph is thus constructed. Then, the model transfer graph is extended to include a target task and a transfer function can be obtained according to a graph theory. Finally, a group of optimal parameters for the target task can be automatically determined by using a least-squares method. Simulation results involving a robot path planning problem show the intelligence, rapidness and reasonability of the proposed method.

Key words: ant colony algorithm; parameters selection; graph knowledge transfer; path planning

1 引 言

从蚂蚁群体寻找最短路径的觅食行为受到启发, Dorigo 等人^[1]于 20 世纪 90 年代提出一种模拟自然界蚁群行为的模拟进化算法——人工蚁群算法. 该算法不仅能智能搜索、全局优化, 而且具有高度的本质并行性、正反馈性、鲁棒性及协同性等优点, 在解决复杂优化问题上显示出良好的适应性, 并已成功应用于解决旅行商^[2]、车间调度^[3]和机器人路径规划^[4]等问题. 但是, 同其他许多启发式智能优化算法一样, 蚁群算法的优化结果对参数的选择比较敏感, 算法运行参数的选取对算法性能有着至关重要的影响^[5-6]. 对于不同的优化问题, 算法的参数选取不同. 即使对于同一类型的优化问题, 由于问题规模不一样, 算法的参

数选取也不尽相同.

传统蚁群算法的参数选取方法主要有: 经验选择法、实验试凑法、正交试验设计法等. 经验选择要求对所研究问题有很好的经验和十足的知识, 否则, 并不容易获得合适的参数. 实验试凑是通过大量的数字仿真实验来获得较优的参数, 比较费时, 而且获得的参数也不一定是最优的. 正交试验设计法的缺点是计算量大, 并且没有全面考虑参数之间存在的耦合性. 由于蚁群算法的参数选取可视为优化问题, 近年来, 陆续有学者采用遗传算法 (GA)^[7]或粒子群优化 (PSO)^[8]等智能优化算法对蚁群算法的运行参数进行迭代求解. 但该方法在进行蚁群算法参数优化的同时, 又引入了诸如 GA 中的选择、交叉、变异概

收稿日期: 2010-08-18; 修回日期: 2010-11-15.

基金项目: 国家自然科学基金项目(60804022, 60974050, 61072094); 教育部新世纪优秀人才支持计划项目(NCET-08-0836); 霍英东教育基金会青年教师基金项目(121066); 江苏省自然科学基金项目(BK2008126).

作者简介: 王雪松(1974—), 女, 教授, 从事机器学习、生物信息学方面的研究; 潘杰(1986—), 男, 博士生, 从事知识迁移学习的研究.

率, PSO 算法中的惯性权值、学习因子、最大速度等众多参数的选择问题, 而 GA 和 PSO 算法的优化性能在很大程度上也取决于其自身运行参数的合理选择.

蚁群算法参数空间的庞大性和各参数之间的耦合性, 使如何确定最优参数组合使蚁群算法的求解性能达到最佳, 成为一个极其复杂的问题. 为解决该问题, 此处考虑采用一种图知识迁移 (GKT) 方法来对蚁群算法的参数进行优化选择. 所谓知识迁移, 就是将以往源任务中已具备的经验知识合理地迁移到目标任务, 使 Agent 能快速地学习新的复杂任务^[9]. 本文方法的主要思路为: 利用图论知识, 通过对各任务 (包括源任务和目标任务) 构造一个模型迁移图来实现知识的迁移. 此处用来迁移的知识是各个源任务的经验参数, 即蚁群算法的运行参数. 具体实现步骤为: 首先, 将源任务映射到一个高维的迁移空间, 并通过迁移权值连接不同的源任务, 以衡量它们之间的迁移性能; 然后, 将模型迁移图进行扩展使其包含目标任务, 并通过 Agent 对迁移函数的学习, 最终为目标任务分配一个优化的迁移参数, 以实现蚁群算法参数的合理选择. 这样, 迁移空间中的任务点以及迁移权值可构成一个模型迁移图. 实际问题中源任务的个数总是有限的, 因而模型迁移图实际上是拥有无数任务点的理想迁移流形的离散化形式, 而有限的源任务点则可看作是 Agent 学习迁移流形的训练样本.

尽管基于图知识迁移的蚁群算法参数选择同样依赖于以往的经验, 但与上述传统的经验选择法相比, 它降低了对执行操作人员的要求. 换言之, 是 Agent 自主地利用过去学得的知识来解决当前问题, 在这个过程中, 不需要人对其进行指导或干预. 同时, 相比于依靠大量数字仿真的实验试凑法, 充分利用以往已掌握的知识显然会降低新任务的学习难度, 使得基于图知识迁移的参数选择算法花费的时间代价更小.

2 蚁群算法参数分析

一般而言, 蚁群算法主要有 5 个运行参数: α , β , ρ , Q , M , 其控制作用体现在以下 3 个式子中^[1,10]:

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{k \in \text{allowed}} [\tau_{ik}(t)]^\alpha [\eta_{ik}]^\beta}, & \text{如果 } j \in \text{allowed}; \\ 0, & \text{否则.} \end{cases} \quad (1)$$

$$\tau_{ij}(t+1) = \rho \tau_{ij}(t) + \Delta \tau_{ij}. \quad (2)$$

$$\Delta \tau_{ij} = \begin{cases} Q/L_k, & \text{如果第 } k \text{ 只蚂蚁经过路段 } ij; \\ 0, & \text{否则.} \end{cases} \quad (3)$$

式 (1) 中: $p_{ij}^k(t)$ 为第 k 只蚂蚁从结点 i 出发选择

结点 j 的概率; allowed 为蚂蚁处于结点 i 时的下一步可行点集, 由环境地图决定; 参数 α 表示路段 ij 上残留信息素 $\tau_{ij}(t)$ 的相对重要程度, 反映了上一代蚂蚁在路径中生成的信息素对当前蚂蚁选择路径的影响, 其值越大, 当前蚂蚁越倾向于选择之前蚂蚁走过的路径; 参数 β 表示结点 j 处启发式因子 η_{ij} 的相对重要程度, 反映了当前蚂蚁对于目标结点的可见度.

式 (2) 给出了信息素浓度的更新规则. 其中: $\Delta \tau_{ij}$ 为时刻 t 到时刻 $t+1$ 的新增信息素, 由式 (3) 求得; 参数 ρ 为信息素的持久性, $1-\rho$ 为信息素轨迹的衰减度, 即信息素的消逝程度. 若 ρ 太大, 算法的全局搜索能力就会降低; 若减小 ρ , 算法的全局搜索能力会随之提高, 但收敛速度会变慢.

式 (3) 中: L_k 为第 k 只蚂蚁所寻路径的长度; 参数 Q 为新增信息素 $\Delta \tau_{ij}$ 的强度, 反映了路径上信息素增加速度的快慢. 较大的 Q 值会使信息素浓度过于集中, 令算法陷入局部最优; 而 Q 值太小又会使算法寻路的速度变慢.

蚁群算法中的参数 M 表示蚂蚁的总数, 它的大小在一定程度上影响 Q 值的选取. 另外, 与参数 ρ 类似, 参数 M 的合理选择同样陷入算法全局搜索能力与收敛速度的矛盾之中.

3 基于图知识迁移的蚁群算法参数选择

3.1 邻接迁移矩阵

给定源任务集合 $T = \{t_i\}_{i=1}^n$, 对于每个指定的源任务 t_i , 采用蚁群算法模型 m_i 进行优化, 以选择最优路径 L_{best} . 每个模型 m_i 有其相应的运行参数组合 $p_i = (\alpha_i, \beta_i, \rho_i, Q_i, M_i)^T \in R^5$, 该参数组合 p_i 可在不同任务 t_i 之间进行迁移. 现将源任务集合 $T = \{t_i\}_{i=1}^n$ 映射到一个特殊的高维空间 R^θ , R^θ 称为源任务的迁移空间. 该空间用于描述不同任务间的迁移性能, 空间的维数 θ 等于迁移参数 p_i 的维数, 即 $\theta = 5$. 迁移空间的每个点 P_i 均代表一个不同的任务 t_i , 各任务 t_i 的坐标用其相应的运行参数 p_i 来表示. 不同任务 t_i 与 t_j 间用迁移权值 $\omega_{i,j}$ 进行连接. 迁移权值 $\omega_{i,j}$ 的定义如下:

定义 1 设任务 t_i 与 t_j 分别有相应蚁群模型 m_i 与 m_j , 及运行参数 p_i 与 p_j , 则迁移权值

$$\omega_{i,j} = (\text{length}_j - \text{length}_{i,j}) / \text{length}_j. \quad (4)$$

其中: $\text{length}_{i,j}$ 表示蚁群模型 m_j 在参数 p_i 控制下的路径长度, length_j 表示 m_j 在自身参数 p_j 控制下的路径长度.

由定义 1 可知, 迁移权值 $\omega_{i,j}$ 实际上表示任务 t_i 对任务 t_j 的参数迁移性能. 需要说明的是 $\omega_{i,j} \neq \omega_{j,i}$, 即不同任务 t_i 与 t_j 间的相互迁移不具有对称性.

由于 $\text{length}_{i,j} \in (0, +\infty)$ 且 $\text{length}_j \in (0, +\infty)$, 使迁移权值 $\omega_{i,j}$ 的取值范围为 $(-\infty, 1)$. 当 $\omega_{i,j} > 0$ 时表示正迁移, 即迁移参数的性能优于原参数的性能, 其值越接近于 1, 迁移效果越好; 当 $\omega_{i,j} < 0$ 时表示负迁移, 即参数迁移后算法性能变差, 其值越小, 迁移效果越差; $\omega_{i,j} = 0$ 时性能不变, 即参数迁移前后对算法没有影响. 迁移权值的总合构成了邻接迁移矩阵 A_n , 定义如下:

定义 2 设高维迁移空间 R^θ 中具有 n 个不同的源任务 $\{t_i\}_{i=1}^n$, 且任务 t_i 对 t_j 的迁移权值为 $\omega_{i,j}$, 则有邻接迁移矩阵 $A_n = \{a_{i,j}\}$, 其中

$$a_{i,j} = \begin{cases} 0, & \text{如果 } i = j; \\ \min(\omega_{i,j}, \omega_{j,i}), & \text{否则.} \end{cases} \quad (5)$$

对于迁移空间 R^θ , 采用模型迁移图 G 对其进行建模^[9]. 该模型迁移图包括两方面内容: 一是由空间中的任务点 t_i 构成的任务点集 $P = \{P_i\}_{i=1}^n$; 二是由迁移权值 $\omega_{i,j}$ 构成的邻接迁移矩阵 A_n , 即 $G = (P, A_n)$.

3.2 扩展的模型迁移图

根据 3.1 节的内容, 在已知路径规划的源任务集合 $T = \{t_i\}_{i=1}^n$ 以及相应的蚁群算法模型参数 $p_i = (\alpha_i, \beta_i, \rho_i, Q_i, M_i)^T$ 的基础上, 能采用模型迁移图 $G = (P, A_n)$ 来描述源任务及其相互间的迁移关系. 本节将说明如何扩展模型迁移图 G , 使其包含目标任务 t_{n+1} , 并通过扩展的模型迁移图 $\hat{G} = (\hat{P}, \hat{A}_{n+1})$ 来优化目标任务 t_{n+1} 的相应参数 $p_{n+1} = (\alpha_{n+1}, \beta_{n+1}, \rho_{n+1}, Q_{n+1}, M_{n+1})^T$.

对模型迁移图 G 进行扩展, 利用式 (4) 计算各源任务 $\{t_i\}_{i=1}^n$ 对目标任务 t_{n+1} 的扩展迁移权值 $\hat{\omega}_1, \hat{\omega}_2, \dots, \hat{\omega}_n$. 扩展的模型迁移图 \hat{G} 定义如下:

定义 3 设 p_{n+1} 为目标任务 t_{n+1} 对应的随机运行参数, $\hat{\omega} = (\hat{\omega}_1, \hat{\omega}_2, \dots, \hat{\omega}_n)$ 为源任务对目标任务的迁移权值, 则扩展模型迁移图 $\hat{G} = (\hat{P}, \hat{A}_{n+1})$, 其中 $\hat{P} = P \cup \{P_{n+1}\}$, P_{n+1} 为迁移空间中对应于目标任务的点, \hat{A}_{n+1} 由下式给出^[9]:

$$\hat{A}_{n+1} = \begin{bmatrix} A_n & \hat{\omega}^T \\ \hat{\omega} & 0 \end{bmatrix}. \quad (6)$$

将模型迁移图 G 扩展为包含目标任务 t_{n+1} 的 $\hat{G} = (\hat{P}, \hat{A}_{n+1})$ 后, 接下来要学习基于图 \hat{G} 的迁移函数. 所谓迁移函数 \hat{f} , 是包括目标任务 t_{n+1} 在内的任务集合 $\hat{T} = \{t_i\}_{i=1}^{n+1}$ 到参数集合的映射, 即 $\hat{f}: \hat{T} \rightarrow R^\theta$, 它给每一个任务 t_i 分配一个针对其蚁群模型 m_i 的运行参数 $p_i = (\alpha_i, \beta_i, \rho_i, Q_i, M_i)^T$. 由于源任务 T 的所有参数信息 $\{p_i\}_{i=1}^n$ 均是已知的, 使其可以作为学习迁移函数的样本. 由这些训练数据 $\{p_i\}_{i=1}^n$ 可生

成一个合适的迁移函数, 它将源任务的参数知识迁移给目标任务 t_{n+1} , 从而给目标任务分配一个优化的运行参数 \hat{p}_{n+1} .

为了使迁移函数 \hat{f} 符合任务间的迁移关系, 需要在模型迁移图 $\hat{G} = (\hat{P}, \hat{A}_{n+1})$ 的基础上建立迁移函数. 为此, 引入图论的知识进行分析求解.

3.3 目标任务的参数优化

设 $G = (P, A_n)$ 为任务集合 T 对应的模型迁移图, 它是包括 n 个顶点与 $n \times n$ 个迁移权值的 θ 维无向连通图. 图上各个顶点 P_i 的度为 $d_i = \sum_{j=1}^n a_{i,j}$, 则度对角矩阵 $D = \text{diag}(d_1, d_2, \dots, d_n)$. 由于模型迁移图 G 与图拉普拉斯矩阵 $L = D - A_n$ 之间存在一一对应的关系, 将 L 进行特征分解, 有 $L = Q\Lambda Q^T$. 其中: Q 为 L 的特征向量矩阵; $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ 为 L 的特征值对角矩阵, 且有 $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$. 这样, 可得到基于图 G 的一组正交基向量矩阵 Q .

以上分析过程均是针对源任务的模型迁移图 G 进行的. 相应地, 针对扩展迁移图 \hat{G} , 有图拉普拉斯算子 \hat{L} , 特征值对角矩阵 $\hat{\Lambda}$ 以及作为基的特征向量矩阵 \hat{Q} .

设迁移函数矩阵 $\hat{f} = (p_1, p_2, \dots, p_{n+1})^T$, 在求得 \hat{Q} 后, 可以得知, 必然存在一个 $(n+1) \times \theta$ 维的权值矩阵 W , 使得 $\hat{f} = \hat{Q}W$. 取 \hat{Q} 的前 n 行作为源任务的一组基, 即 $Q = \hat{Q}_{1,2,\dots,n,*}$, 则采用最小二乘法, 可以得到如下权值矩阵^[9]:

$$W_{*,i} = \arg \min_{\omega} (\|f_{*,i} - Q\omega\|^2 + \|\sqrt{\hat{\Lambda}}\omega\|^2). \quad (7)$$

其中: $f = (p_1, p_2, \dots, p_n)^T$, $\|\sqrt{\hat{\Lambda}}\omega\|^2$ 为基于 f 的二阶导数惩罚项. 为防止过拟合现象的出现, 惩罚项可以很好地抑制 f 中的高频成分, 其推导过程如下^[9]:

$$\begin{aligned} \langle \nabla \hat{f}, \nabla \hat{f} \rangle &= \langle \hat{f}, \hat{L} \hat{f} \rangle = (\hat{Q}W)^T (\hat{L} \hat{Q}W) = \\ \omega^T \hat{Q}^T (\hat{Q} \hat{\Lambda} \hat{Q}^T \hat{Q}W) &= \omega^T I \hat{\Lambda} I \omega = \\ \omega^T \hat{\Lambda} \omega &= \|\sqrt{\hat{\Lambda}}\omega\|^2. \end{aligned} \quad (8)$$

对式 (7) 进行求解可得

$$W = (Q^T Q + \hat{\Lambda})^{-1} Q^T f. \quad (9)$$

求出权值矩阵 W 后, 再利用迁移函数 $\hat{f} = \hat{Q}W$ 求得 \hat{f} . 则该矩阵的第 $n+1$ 行转置后即为目标任务的优化迁移参数 \hat{p}_{n+1} .

4 仿真研究

为了对本文所提蚁群算法的参数选择方法进行深入的分析研究, 考虑采用机器人路径规划的任务进行仿真. 图 1 给出了仿真的 6 幅地形图, 地图中无色格子代表可通行区域, 有色格子代表障碍物. 地

图尺寸大小及障碍物覆盖率分别为: map 1: 20×20, 28.5%; map 2: 20×20, 34.5%; map 3: 20×20, 29.3%; map 4: 20×20, 29.0%; map 5: 16×16, 34.8%; map 6: 40×40, 39.4%. 其中: 图 1(a)~图 1(e)对应源任务 t_i , $i = 1, 2, \dots, 5$, 图 1(f)对应目标任务 t_6 . 各任务的目的地相同, 均要求机器人能够快速找到一条从出发点到达目的地的最短路径, 出发点设为地图左上角, 目的地设为右下角. 机器人在地图中有 8 个移动方向, 即前、后、左、右、左前、右前、左后和右后, 每次只能移动一个单元格, 在碰到边沿或障碍物时停在原地. 仿真的目标是合理地将 5 个源任务已具有的参数知识迁移到目标任务中, 为其分配一个优化参数, 以期快速规划出一条最短路径.

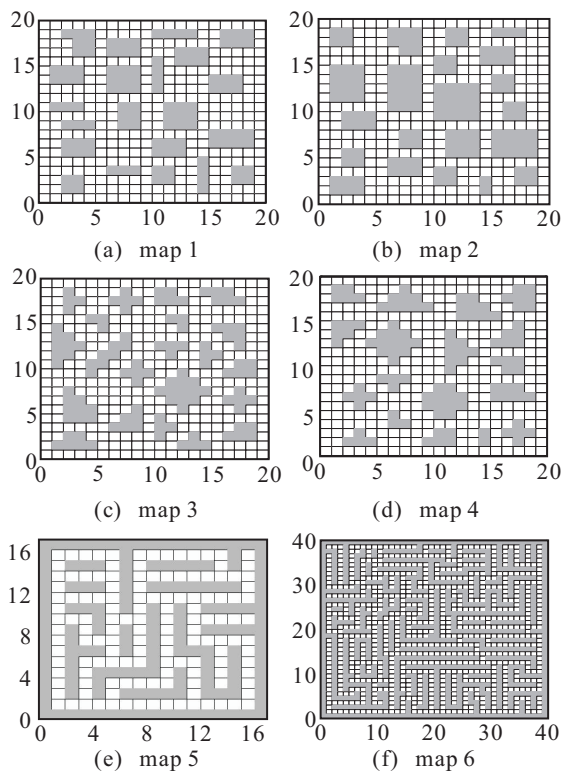


图 1 源任务与目标任务地图

已知各源任务的经验参数为: map 1: $\alpha = 5, \beta = 10, \rho = 0.3, Q = 1, M = 5$; map 2: $\alpha = 10, \beta = 8, \rho = 0.2, Q = 0.08, M = 10$; map 3: $\alpha = 6, \beta = 10, \rho = 0.1, Q = 0.1, M = 30$; map 4: $\alpha = 6, \beta = 4, \rho = 0.5, Q = 1, M = 40$; map 5: $\alpha = 2, \beta = 10, \rho = 0.2, Q = 1, M = 5$. 不同源任务间的相互迁移性能如表 1 所示.

表 1 中, 行对应源任务, 列对应被迁移的任务, 迁

表 1 源任务间的迁移性能

	map 1	map 2	map 3	map 4	map 5
map 1	/	-0.108 2	-0.010 3	-0.028 5	-0.106 0
map 2	0.130 4	/	-0.022 3	-0.078 0	0.017 0
map 3	0.066 0	0.010 9	/	0.046 8	-0.058 0
map 4	0.165 5	-0.108 2	-0.196 9	/	0.042 7
map 5	0.115 7	-0.044 8	-0.035 0	0.062 4	/

移性能由迁移权值 $\omega_{i,j}$ 来衡量, 其值由式 (4) 计算得出. 为消除蚁群算法的随机性对衡量参数性能时的不良影响, 取 10 次仿真时的平均值. 从表 1 中可以直观地看出, 不同任务间的相互迁移不具有对称性. 另外, 尽管各个任务采用的均是令人满意的经验参数, 但仍然有优劣之分. 对比第 1 列与第 3 列的数据可以发现, 各任务对 t_1 均有正的迁移效果, 而 t_3 在接受其他任务的参数迁移时则显示出负的迁移特性, 这说明任务 t_1 的经验参数略差, 而 t_3 的经验参数则较优. 观察第 3 行数据, 除了任务 t_3 对任务 t_5 的迁移效果为 -0.0580 外, 其余均为正迁移, 这也说明任务 t_3 的经验参数在几个源任务中, 具有一定的普适性.

表 2 给出了各源任务对目标任务 t_6 的迁移性能, 为衡量这种性能的优劣, 对任务 t_6 分配随机运行参数, 同样取 10 次仿真的平均值作为表中的数据. 由此可以看出, 在源任务中 t_3 对目标任务具有最佳迁移性能, 因此可将该任务点作为下面进行最佳手动迁移时的依据. 另外, 尽管从地形图来看, map 5 与 map 6 具有一定的相似性, 但这并不能说明任务 t_5 对目标任务 t_6 拥有最好的迁移效果.

表 2 源任务对目标任务的迁移性能

目标任务	map 1	map 2	map 3	map 4	map 5
map 6	0.227 7	0.057 8	0.333 7	0.005 5	0.193 8

为考察基于图知识迁移算法的性能优劣, 将其与其他类型的知识迁移算法进行对比, 仿真效果如图 2 所示. 这里, 仅取路径长度作为评价标准, 因为此处衡量迁移性能的唯一依据就是路径长度. 若要考虑其他因素, 如路径规划总耗时及算法迭代次数等, 须对定义 1 及式 (4) 进行修正.

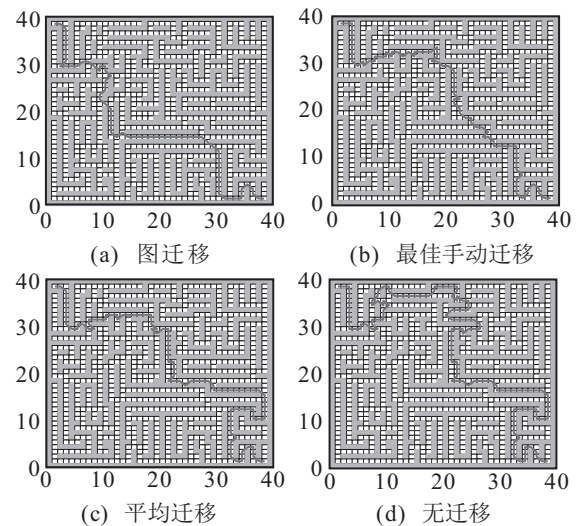


图 2 各类迁移算法规划路径效果对比

图 2(a) 为基于图的知识迁移算法 GKT, 其规划路径长度为 80.284 3, 为各类迁移算法中最优. 图 2(a) 中, $\alpha = 5.404 3, \beta = 7.471 7, \rho = 0.356 7, Q = 2.124 9,$

$M = 25$. 图 2(b) 为最佳手动迁移 (BMKT) 算法得到的优化路径, 长度为 89.3553. 图 2(b) 中, $\alpha = 6$, $\beta = 10$, $\rho = 0.1$, $Q = 0.1$, $M = 30$. 所谓最佳手动迁移, 即将源任务中对目标任务迁移性能最好的参数迁移到目标任务, 其参考依据为表 2 中数据, 本实验取任务 t_3 的经验参数进行迁移. 平均迁移 (AKT) 为简单地将源任务中各经验参数取平均值后迁移到目标任务中. 图 2(c) 给出其优化蚁群参数后的规划路径, 长度为 107.4558. 图 2(c) 中, $\alpha = 5.8$, $\beta = 8.4$, $\rho = 0.26$, $Q = 0.6360$, $M = 18$. 图 2(d) 为无迁移 (NKT) 的规划效果, 即在没有任何经验知识的前提下, 为目标任务分配一组随机运行参数得到的结果. 图 2(d) 中, $\alpha = 9.4083$, $\beta = 7.5914$, $\rho = 0.6257$, $Q = 8.6985$, $M = 38$. 尽管目标任务有极小的概率随机找到较优的参数组合, 但是很显然, 在大多数情况下, 该方法无法得到理想的规划路径.

图 2 仅给出了上述各类迁移算法某一次仿真的路径规划结果. 事实上, 对于蚁群算法而言, 即使是同一组运行参数, 在多次实验中, 仿真结果仍会在小范围中浮动. 为消除算法随机性的影响, 独立进行 10 次仿真, 图 3 为各算法性能对比曲线. 从图中可以看出, 算法 GKT 与 BMKT 的规划路径最短, 性能最佳, 而两者相比, 本文提出的 GKT 算法略优. 至于 AKT 算法, 由于迁移参数仅是对源任务各参数的简单平均, 对目标任务并不具有针对性, 规划路径效果较差. 但与 NKT 算法相比, AKT 算法拥有先验知识, 其迁移参数就路径规划问题本身而言, 具有一定合理性. 而性

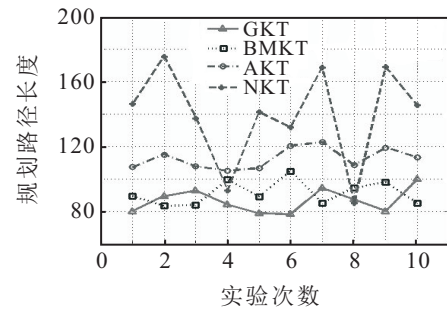


图 3 各类迁移算法 10 次仿真性能对比

能最差的 NKT 算法则是完全盲目地随机分配运行参数. 需要说明的是, 尽管如此, NKT 算法仍有两次随机找到了较优参数.

以上分析的是 GKT 算法与其他类型迁移算法的性能对比, 从已有知识合理迁移与利用的角度说明了算法的有效性. 作为一种智能参数选择方法, 将本文方法与基于 GA^[7] 和基于 PSO^[8] 算法的蚁群参数选择算法进行对比. 表 3 给出了各算法得到的优化参数组合, 以及该参数组合所需的优化时间. 针对得到的各组优化参数, 分别作 10 次路径规划实验, 统计平均路径长度. PSO 算法的参数设定为: 学习因子 $c_1 = c_2 = 2.05$, 惯性权值 $w = 0.5(T_{\max} - t)/T_{\max} + 0.4$. 其中: T_{\max} 为算法最大迭代次数, 取为 100; t 为算法当前运行代数; 粒子个数 $\text{Num} = 10$. GA 的参数设定为: 染色体长度 $L = 20$; 编码精度 $p = 4$, 即采用 4 个二进制位分别对各蚁群参数进行编码; 种群个体数 $N = 10$; 最大进化代数 $K = 100$; 代沟 $P_G = 0.9$; 单点交叉概率 $P_C = 0.7$; 变异概率 $P_N = 0.7/L$.

表 3 各优化算法对蚁群参数的优化性能对比

优化算法	优化参数组合					参数优化时间/s	规划路径平均长度
	α	β	ρ	Q	M		
GKT	5.4043	7.4717	0.3567	2.1249	25	58.5940	86.7127
GA ^[7]	2.8667	13.1333	0.0753	5.3800	31	6.1028×10^3	82.5044
PSO ^[8]	6.0728	11.5516	0.1542	0.7228	48	1.1670×10^4	82.5323

观察表 3 中数据可知, 蚁群算法的优化参数并不唯一, GKT 算法与 GA 和 PSO 算法得到的优化组合不同, 但同样有令人满意的规划效果. 从获得参数的时间来看, GA 与 PSO 算法分别为 6.1028×10^3 s 与 1.1670×10^4 s, 远远高于本文 GKT 算法的优化时间 58.5940 s. 事实上, 虽然 GA 与 PSO 算法是两类不同的进化算法, 但在优化参数的过程中, 均需要反复调用蚁群任务对其个体进行评价, 以使其相应的种群向着更好的方向发展, 该过程将花费大量时间. 而 GKT 算法已具有经验知识, 只需合理迁移即可, 因而能快速获得优化参数. 当然, 作为其快速获得参数的代价, 其所得参数的规划精度略低于另外两种智能

算法. 这是由于 GA 与 PSO 算法对参数的优化不依赖于任务本身, 自身具有独立性; 而 GKT 算法性能的好坏则在一定程度上依赖于源任务与目标任务间的迁移关系.

5 结 论

蚁群算法的优化性能在很大程度上取决于其运行参数的合理设置, 而参数空间的庞大性和参数间的耦合性使蚁群算法的参数选择成为一个既有实际价值, 又有理论意义的研究课题. 本文提出一种基于图知识迁移的蚁群算法参数选择方法. 概括来说, 该方法具有以下 3 个特点:

1) 智能性. 图知识迁移方法能将源任务的参数

知识自动地迁移到目标任务,得到针对目标任务的最优参数组合.由图2和图3可以看出,GKT算法所得运行参数甚至比人为选取源任务中对目标任务迁移最好的参数性能略优.这种将原有知识提炼并迁移的能力,是算法智能性的明显体现.

2) 快速性.与其他智能优化算法相比,拥有经验知识使GKT算法能够快速得到优化参数.尽管本文仅考虑离线的情形,但适当改进即可用于在线的路径规划任务,同时这也是后续研究工作.而对于GA与PSO算法而言,数个小时的优化时间成为其应用于在线任务的巨大障碍.

3) 合理性.不同于现有的绝大多数参数选择方法,本文方法同时考虑蚁群算法的所有运行参数,知识迁移的过程中蕴含了各参数相互间的复杂耦合关系,而并非是对其中一个或某几个参数进行剥离式的单独研究.本文算法充分考虑了各参数的整体配合性.

参考文献(References)

- [1] Dorigo M, Maniezzo V, Colormi A. Ant system: Optimization by a colony of cooperating agents[J]. IEEE Trans on Systems, Man and Cybernetics, 1996, 26(1): 29-41.
- [2] Rais H M, Othman Z A, Hamdan A R. Improved dynamic ant colony system (DACS) on symmetric traveling salesman problem[C]. Proc of the Int Conf on Intelligent and Advanced Systems. Piscataway: IEEE Inc, 2008: 43-48.
- [3] Venkataramana M, Raghavan N R S. Ant colony-based algorithms for scheduling parallel batch processors with incompatible job families[J]. Int J of Mathematics in Operational Research, 2010, 2(1): 73-98.
- [4] Garcia M A P, Montiel O, Castillo O, et al. Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation[J]. Applied Soft Computing, 2009, 9(3): 1102-1110.
- [5] 潘全科, 王凌, 高亮. 离散微粒群优化算法的研究进展[J]. 控制与决策, 2009, 24(10): 1441-1449. (Pan Q K, Wang L, Gao L. The state-of-art of discrete particle swarm optimization algorithms[J]. Control and Decision, 2009, 24(10): 1441-1449.)
- [6] 纪震, 周家锐, 廖惠连, 等. 智能单粒子优化算法[J]. 计算机学报, 2010, 33(3): 556-561. (Ji Z, Zhou J R, Liao H L, et al. A novel intelligent single particle optimizer[J]. Chinese J of Computers, 2010, 33(3): 556-561.)
- [7] Zaitar R A, Hiyassat H. Optimizing the ant colony optimization using standard genetic algorithm[C]. Proc of the 23rd IASTED Int Conf on Artificial Intelligence and Applications. Calgary: ACTA Press, 2005: 130-134.
- [8] Zhou Z G. An improved ant colony optimization supervised by PSO[J]. Advanced Materials Research, 2010, 108-111(1): 1354-1359.
- [9] Eric E, Marie D J, Terran L. Modeling transfer relationships between learning tasks for improved inductive transfer[J]. Lectures Notes in Artificial Intelligence, 2008, 5211: 317-332.
- [10] Zhao D B, Yi J Q. Robot planning with ant colony optimization algorithms[C]. Proc of the 25th Chinese Control Conf. Piscataway: IEEE Inc, 2006: 1460-1465.
- [7] Shi X H, Liang Y C, Lee H P, et al. An improved GA and a novel PSO-GA-based hybrid algorithm[J]. Information Processing Letters, 2005, 93(5): 255-261.
- [8] 方伟, 孙俊, 须文波. 一种多样性控制的粒子群优化算法[J]. 控制与决策, 2008, 23(8): 863-868. (Fang W, Sun J, Xu W B. Diversity-controlled particle swarm optimization algorithm[J]. Control and Decision, 2008, 23(8): 863-868.)
- [9] Trelea I C. The particle swarm optimization algorithm: Convergence analysis and parameter selection[J]. Information Process Letters, 2003, 85(6): 317-325.
- [10] Solis F, Wets R. Minimization by random search techniques[J]. Mathematics of Operations Research, 1981, 6(1): 19-30.
- [11] Clerc M, Kennedy J. The particle swarm: Explosion stability and convergence in a multi-dimensional complex space[J]. IEEE Trans on Evolution Computer, 2002, 6(1): 58-73.
- [12] Eberhart R C, Shi Y. Comparing inertia weights and constriction factors in particle swarm optimization[C]. Proc of the IEEE Congress on Evolutionary Computation. California: IEEE, 2000: 84-88.
- [13] 王建林, 薛尧予, 于涛, 等. 基于群能量恒定的粒子群优化算法[J]. 控制与决策, 2010, 25(2): 269-272. (Wang J L, Xue Y Y, Yu T, et al. Particle swarm optimization based on swarm energy conservation[J]. Control and Decision, 2010, 25(2): 269-272.)

(上接第1854页)