第 30 卷 第 6 期
2004 年 11 月

自 动 化 学 报
ACTA AUTOMATICA SINICA

Vol.30, No.6
Nov., 2004

求解复杂多阶段决策问题的动态窗口蚁群优化算法[1)]

闻 育 吴铁军

(浙江大学智能系统与决策研究所 杭州 310027)

(E-mail: ywen@iipc.zju.edu.cn, tjwu@iipc.zju.edu.cn)

摘 要 针对存在强非线性、系统状态与控制输入复杂约束和非解析系统表达，以及目标函数具有可加性和单调性的大规模多阶段决策问题，提出一种结合遗传优化的动态窗口蚁群优化算法. 该算法将各阶段容许决策值映射为一个层状构造图中的有限节点集，其中每一层节点对应一个阶段的容许决策集合的子集，该子集用实数编码遗传优化进行动态筛选，以减小算法的搜索空间. 经原理分析和仿真比较，该算法的计算效率比一般蚁群算法大大增强.

关键词 多阶段决策，蚁群算法，复杂系统优化

中图分类号 TP202

Dynamic-window-search Ant Colony Optimization for Complex Multi-stage Decision Making Problems

WEN Yu WU Tie-Jun

(*Intelligent Systems & Decision Making Institute Zhejiang University, Hangzhou* 310027)

(E-mail: ywen@iipc.zju.edu.cn, tjwu@iipc.zju.edu.cn)

Abstract A dynamic-window-search ant colony optimization (ACO) algorithm, integrated with genetic optimization techniques, is proposed for large-scale multi-stage decision making problems, which are of strong nonlinearity, complex constraints on system states and control inputs, non-analytical system representation, and additive and monotonic objective functions. A subset of the feasible decision set at each stage is dynamically selected for the algorithm by real-coded genetic optimization and is mapped to the nodes of the corresponding layer in a layered construction graph to reduce the size of the search space. Computational complexity analysis and simulation results demonstrate that, in comparison with basic ACO algorithms, the proposed algorithm greatly improves the computational efficiency.

Key words Multi-stage decision making problem, ant colony optimization, complex systems optimization

1) 国家技术研究与发展计划项目 (2002AA412010) 资助
Supported by National High Thechnolgy Research and Development Program (2002AA412010)
收稿日期 2003-07-14 收修改稿日期 2003-11-24
Received July 14, 2003; in revised form November 24, 2003

1 引言

多阶段决策问题的数学模型可定义为[1]

$$\min \{J(N) = F[\boldsymbol{x}(0), s_{0,N}]\} \tag{1}$$

$$\text{s.t. } \boldsymbol{x}(t+1) = f[\boldsymbol{x}(0), s_{0\ t+1}] \tag{2}$$

$$s_{0,t+1} = \{\boldsymbol{u}(0), \boldsymbol{u}(1), \cdots, \boldsymbol{u}(t)\} \tag{3}$$

$$\boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{4}$$

$$\boldsymbol{x}(t) \in X(t) \subseteq R^n \tag{5}$$

$$\boldsymbol{u}(t) \in U(t) \subseteq R^r \tag{6}$$

$$t = 0, 1, \cdots, N-1$$

上式中 $J(N)$ 为目标函数值；F 为目标函数；$\boldsymbol{x}(t)$ 为状态变量；$X(t)$ 为容许状态集合；$\boldsymbol{u}(t)$ 为决策变量；$U(t)$ 为容许决策集合；$s_{0\ N}$ 为问题的一个全过程策略，简称策略，$s_{0,t-1}$ 为其子策略；方程 (2) 称为状态转移方程，f 为状态转移函数，$t+1$ 阶段的状态是初始状态 $\boldsymbol{x}_0$ 和子策略 $s_{0,t+1}$ 决定的. 当 F 或 f 具有强非线性或难以解析表达，X 或 U 为高维复杂拓扑空间时，称其为复杂多阶段动态决策问题.

大量的复杂系统优化决策和最优控制问题都可以用上述模型来描述，如计算机广域网路由控制问题、柔性制造系统生产调度问题、智能交通系统协调优化问题等等，由于问题自身的复杂性，使得经典的求解方法难以实际应用. 以城市交通网络的信号灯协调控制问题为例，通常要求优化未来一段时间内各交叉口的信号灯相位序列使一个交通性能指标 (如停车时间) 达到最优. 交通系统的状态转移方程和约束条件都很复杂，具有时滞和非线性，有些交通流模型还没有解析形式. 此外，其决策向量的维数等于交通系统中受控交叉口的个数，若同时控制成百上千个交叉口，解空间非常庞大[2,3].

满足最优化原理和无后效性的多阶段决策问题可以用动态规划求解[1]. 但对于非线性系统或非二次型目标函数问题，通常无法推导出解析形式的递推方程，要用基本递推公式做数值计算. 然而该数值算法的一个明显弱点是，随着状态变量与决策变量维数的增加，计算量和存储量呈指数增长[4].

遗传算法 (GA) 也常被应用于求解复杂多阶段动态决策问题. 不过在应用中存在两个问题：一是对所求解问题的约束条件的描述能力较差，目前所用的方法都存在一定局限性[5]；二是仅依赖解的适应度值来引导对解空间的搜索，对于各阶段决策变量之间具有较强的关联性的问题，无法利用这些关联性信息来提高搜索的效率.

蚁群算法是对自然界中蚂蚁的食物搜索行为的一种仿真，通过将问题的解分解成一些组成元素，由蚂蚁在一个构造图上以随机方式逐步选择各个组成元素构造出问题的解，并将对解的评价结果以信息素的形式释放到连接上，从而引导了其他蚂蚁的搜索，这一过程本质上同样可视为一个多阶段决策过程. 因此，用蚁群算法求解多阶段决策问题十分自然，问题的解本身就是一个决策序列，可以直接让蚂蚁逐步选择各阶段的决策值即构成解. 本文主要研究用蚁群优化算法求解具有目标函数可加性和单调性的复杂多阶段动态

决策问题. 目标函数的形式为

$$J(N)=\sum_{t=0}^{N-1}L[\boldsymbol{x}(t),\boldsymbol{u}(t),t]\quad L[\boldsymbol{x}(t),\boldsymbol{u}(t),t]\geqslant 0 \tag{7}$$

其中 L 为各个阶段的独立目标函数. 对于这类问题, 在解的构造过程中可以充分利用各阶段的目标函数值的增量来引导解的构造方式 (也即搜索方向), 提高搜索的效率. 而对于各类约束条件, 令蚂蚁在各阶段放弃导致非法解的决策值即可.

求解具有高维决策变量的多阶段决策问题会遇到各阶段离散化容许决策集合规模指数膨胀的问题, 为解决这一问题, 我们在迭代过程中通过遗传优化从容许决策集合 $U(t)$ 中抽取一个 "合适" 的子集, 并进行节点映射, 从而提出一种可以动态调整各阶段容许决策集合中的搜索窗口的动态窗口蚁群优化算法.

文中对该算法的计算效率进行了分析, 并以一个多变量非线性时变的最优控制问题为例, 对一般蚁群优化算法和动态窗口蚁群优化算法进行了仿真比较.

2 多阶段动态决策的蚁群搜索

2.1 蚁群算法的基本思想

可以用一个构造图 $G(V,E)$ 来表示蚁群算法所解决的优化问题, 其中 V 是节点的有限集合, E 是有向连接的有限集合. 在构造图中, 节点对应解的组成元素, 一个节点序列 (即路径) 对应问题的一个解.

蚂蚁在路过每一个节点时都会根据与该节点相连的各个连接上的信息素及其所对应的局部启发信息来选择下一个连接, 其公式如下[6,7]:

$$p_{ij}^{a}(k)=\begin{cases}\dfrac{[\tau_{ij}(k)]^{\alpha}[\eta_{ij}^{a}(k)]^{\beta}}{\sum_{l\in A_a(i)}[\tau_{il}(k)]^{\alpha}[\eta_{il}^{a}(k)]^{\beta}}, & \text{if}\quad j\in A_a(i)\\ 0, & \text{if}\quad j\in A_a(i)\end{cases} \tag{8}$$

这里 $p_{ij}^{a}(k)$ 是蚂蚁 a 在第 k 次迭代时选择节点 i 与节点 j 之间连接的概率; $A_a(i)$ 是从节点 i 所能够到达的节点集合, 蚂蚁 a 在节点 i 处选择这些节点的概率之和为 1; $\tau_{ij}(k)$ 为连接 (i,j) 上的信息素; $\eta_{ij}^{a}(k)$ 为局部启发信息, 上标 a 表明与蚂蚁 a 所走过的路径是相关的; $\tau_{ij}(k)$ 和 $\eta_{ij}^{a}(k)$ 是引导蚂蚁构造问题解的两种启发信息, $\tau_{ij}(k)$ 是路过该连接的蚂蚁在完成一条路径后, 根据对整体解的评价留下的信息素, 而 $\eta_{ij}^{a}(k)$ 仅反映当前状态下选择该连接的优劣, 不考虑后面的路径和其他蚂蚁的经验; α,β 是协调上述两种因素的系数, 其取值原则见文献 [5], 一般而言, 对于适用贪婪法则的问题, β 应较大, 要加快收敛, α 应较大.

在蚁群算法中, 蚂蚁从初始节点出发, 通过重复在源节点 (当前所在节点) 处选择下一个要到达的目标节点的操作, 搜索到一条可行路径, 该路径对应问题的一个可行解. 蚂蚁在每个节点处都要随机选择下一个要到达的节点, 这可以看作是一个随机决策, 而目标节点则对应一个决策值. 所以, 蚂蚁是通过一个多阶段决策过程来搜索一条路径, 而所形成的路径实质上就是一个决策序列 (问题解, 或称策略).

2.2 多阶段动态决策的蚁群搜索

在采用蚁群搜索来求解目标函数具有可加性和单调性的多阶段决策问题时，首先将每一阶段的决策变量 $\boldsymbol{u}(t)$ 在其容许决策集合范围内离散化为一系列离散格点 $\boldsymbol{u}_l(t) \in U(t), l = 1, 2, \cdots, q^r$ (设 r 维决策向量的每一维都均匀离散为 q 个格点), 称为离散化决策变量. 记阶段 t 所有决策变量离散格点集合为 $U_d(t) \subset U(t)$, 称为离散化容许决策集合. 然后将各阶段的离散化决策变量 $\boldsymbol{u}_l(t)$ 的取值与图 G 中的节点做一对一的映射，可有

$$\psi : \boldsymbol{u}_l(t) \in U_d(t) \leftrightarrow v_l^t \in V^t \tag{9}$$

这个映射导出了如图 1 所示的层状图，我们称其中每一水平层的节点为决策层，对应某阶段离散化容许决策集合. V_t 为第 t 层的节点集合，v_l^t 为该层的第 l 个节点. 在该层状图中，蚂蚁从初始节点 v_0 出发，重复地选择上一层某个节点，最终形成一条路径 $\{v_0, v_n(0), \cdots, v_{n(t)}, \cdots, v_{n(N-1)}\}$, $n(t)$ 为第 t 阶段选择的节点序号，该路径对应一个可行解.

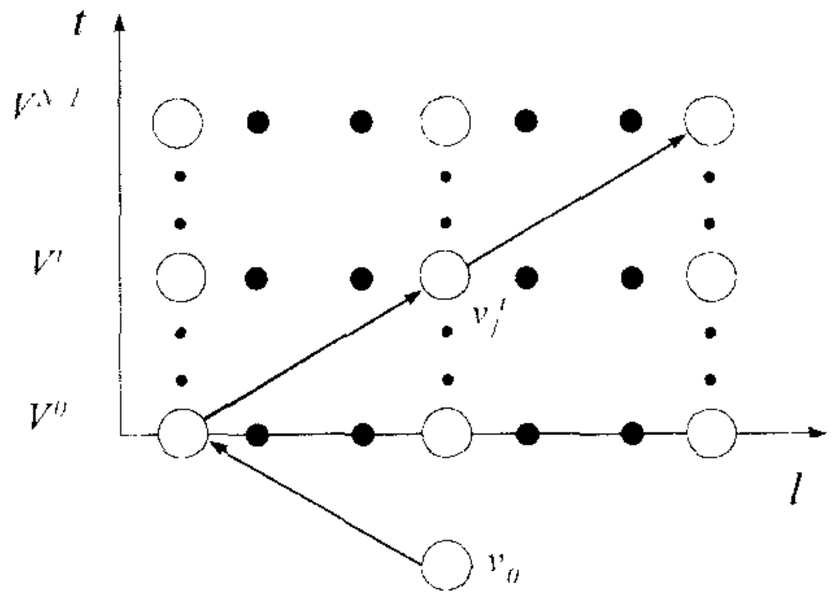

图 1 层状构造图

Fig. 1 Layered construction graph

由于蚂蚁是逐个阶段地选择决策值以构造问题的解，故只需定义相邻层之间的连接，即 (v_i^t, v_j^{t+1}), 并定义连接的代价为

$$\begin{aligned} Cost(v_i^t, v_j^{t+1}) =& [J(t+1) - J(t)] + P(t+1) = \\ & L[x(x+1), u(t+1), t+1] + P(t+1) = \\ & \begin{cases} \infty, & P(t+1) = \infty \\ L[x(t+1), u(t+1), t+1], & P(t+1) = 0 \end{cases} \end{aligned} \tag{10}$$

其中 $P(t+1)$ 为一个由约束条件决定的函数，如果解不满足优化模型中的某个约束条件，则令 $P(t+1) = \infty$, 阻止蚂蚁选择该连接；否则 $P(t+1) = 0$. 连接的代价函数的第一部分等于选择节点 v_j^{t+1} 后所增加的目标函数值. 方程 (2) 表明 $\boldsymbol{x}(t+1)$ 的值决定于初始状态 $\boldsymbol{x}_0$ 和子策略 $s_{0,t+1}$, 而该子策略对应层状图中从初始节点 v_0 至节点 v_i^t 的一条路径，故连接 (v_i^t, v_j^{t+1}) 的代价只能在蚂蚁到达节点 v_i^t 后确定. 连接 (v_i^t, v_j^{t+1}) 上的局部启发信息 $\eta_{ij}^a(k)$ 可定义为连接代价 $Cost(v_i^t, v_j^{t+1})$ 的倒数.

图 1 中，如各阶段离散化容许决策集合的规模为 q^r, 则所形成的层状构造图中每一层的节点总数就为 q^r, 层状图规模随着 r 呈指数增长. 同时，蚂蚁在每一层都要对下一层所有节点进行选择，构造一个解的计算量也呈指数增长.

3 动态窗口搜索算法

3.1 搜索窗口的产生和更新

对于具有高维决策变量的多阶段决策问题，其各阶段离散化容许决策集合的规模会“指数膨胀”，让蚂蚁每个决策阶段在整个离散化容许决策集合中进行选择是困难的. 因此，我们不再将各阶段的决策变量离散化并全部映射为层状图中的节点，而只按某种优化规则 (本文用遗传优化) 从容许决策集合 $U(t)$ 抽取决策变量 $\boldsymbol{u}(t)$ 的 w 个不同取值，进行节点映射. 称由这 w 个决策变量取值构成的集合 $U_s(t) \subset U(t)$ 为蚂蚁在容许决策集 $U(t)$ 中的搜索窗口，$U_s(t)$ 的元素个数 w 为搜索窗口宽度. 这样，蚂蚁只在每一阶段的搜索窗口 $U_s(t)$ 的 w 个节点上进行路径搜索.

在蚁群搜索的构造图中，称所有指向某节点的有向连接为该节点的上游连接. 由方程 (8) 可知，一个节点的上游连接上的信息素越多，其被蚂蚁选中构造新解的概率就越大. 由于信息素还反映了相应决策值所构成解的优劣，这也表明在该阶段选择该决策值构成的解“较有希望”成为最优解.

对于连续的决策变量，可采用实数编码遗传优化来更新搜索窗口. 在若干次蚁群搜索迭代后，首先确定搜索窗口 $U_s(t)$ 中每个节点 (也即其所对应的决策值) 的适应度值. 根据上述分析，可定义节点适应度值 G 为其上游连接上的信息素的平均值

$$G[\boldsymbol{u}_l(t)] = G(v_l^t) = \frac{1}{N_v} \sum_{v \in V^{t-1}} \tau(v, v_l^t) \tag{11}$$

其中 N_v 为节点 v_l^t 上游连接的总数，$\tau(v, v_l^t)$ 为节点 v_l^t 上游连接上的信息素值，v 属于层状图中 $U_s(t-1)$ 所映射的决策层 V^{t-1}.

然后，对搜索窗口 $U_s(t)$ 中的元素进行选择、交叉以及变异的遗传操作，用得到的下一代决策值更新原有的搜索窗口，并再赋给相应决策层中的节点. 因为原节点所对应的决策值发生了变化，所以要重新定义节点的上游连接的信息素值. 又由于节点新对应的决策值是通过上一代的两个父值经交叉和变异操作产生的，可定义新的上游连接信息素值为两个父值的适应度的几何平均

$$\tau_c(v, v_l^t)|_{v \in V^{t-1}} = (d_{cp_1} G_{p_1} + d_{cp_2} G_{p_2})/(d_{cp_1} + d_{cp_2}) \tag{12}$$

这里 $\tau_c(v, v_l^t)$ 为节点 v_l^t 上游连接的新信息素值；G_{p_1} 与 G_{p_2} 分别为节点 v_l^t 所对应的新决策值在遗传操作中的两个上一代决策值 (父值) 的适应度值，其值由方程 (11) 定义；d_{cp_1} 与 d_{cp_2} 为该节点所对应的决策值与这两个父值在解空间中的距离，反映了该决策值对两个父值中的模式的继承程度，这也是对前一次迭代所产生的信息素进行了再分配.

由于遗传操作在若干次蚁群迭代后根据连接上的信息素分布情况调整蚂蚁在各个阶段的搜索窗口，一方面使蚂蚁在构造解时避免了对各个阶段的所有决策值进行比较，而是根据方程 (11) 和遗传优化原理在各阶段的决策域中进行优选，可提高搜索效率；另一方面根据方程 (12) 重新分配连接的信息素，使信息素不只集中节点的某一条上游连接上，而是在节点的所有上游连接上进行传播. 因此，当 $\tau_c(v, v_l^t)$ 较大时，到达上游节点 $v(v \in V_{(t+1)})$ 的蚂蚁选择节点 v_l^t 的可能性都较大. 一般蚁群算法由于信息素的正反馈作用使某一条路

径上的信息素远远超过其他路径，虽然有利于收敛，但也易引起“停滞”问题，这对搜索全局最优解不利. 根据上述方式重新分配连接信息素，可有效地解决“停滞”问题.

3.2　算法基本步骤

步骤 1. 初始化. 抽取每一个阶段的可行决策值构成搜索窗口，并视为遗传优化的一个群落. 根据方程 (9) 将搜索窗口中的决策变量值映射为蚁群搜索层状图中的相应层节点.

步骤 2. 进行若干次蚁群搜索迭代. 蚁群通过层状图生成可行解，并根据对解的评价，更新层状图中各连接上的信息素值，更新公式见文献 [7].

步骤 3. 进行若干次遗传操作，更新各阶段的搜索窗口. 根据方程 (11) 计算原搜索窗口中各节点所对应决策值的适应度值. 在各阶段搜索窗口中运用遗传操作生成新一代的决策值，更新搜索窗口中的节点，并根据方程 (12) 初始化这些节点的上游连接的信息素. 回到步骤 2.

3.3　算法的计算效率分析

蚁群搜索的计算量包括两个方面：一是所有蚂蚁的总迭代次数，若有 M 个蚂蚁各迭代了 C 次，则总次数为 CM 次，迭代次数 C 可以由外部指定或由算法自己决定；二是每只蚂蚁通过构造图生成一个解所要进行的计算量，这主要反映在蚂蚁每次选择下一阶段的决策值都要计算通向该阶段所映射节点的各条连接的选择概率 p_{ij}，故蚂蚁构造一个解的计算量为 $O\left(\Sigma_{t=0}^{N-1} n_t\right)$，其中 n_t 为第 t 层节点的个数. 所以全部计算量为 $O\left(CM\Sigma_{t=0}^{N-1} n_t\right)$.

采用动态窗口后，设各阶段的搜索窗口宽度均为 w(等于遗传优化中的种群规模)，并且每隔 T 次蚁群迭代 (计算量为 $O\left(TM\Sigma_{t=0}^{N-1} n_t\right)$)，对搜索窗口 $U_s(t)$ 进行一次遗传迭代更新，这需进行 $w/2$ 次遗传操作来更新 $U_s(t)$ 中的所有元素，更新 N 阶段搜索窗口的计算量共为 $O(Nw^2)$.

表 1 对两种算法各迭代 C 次的计算量进行了比较. 这里设各阶段的离散化容许决策集合的规模为 q^r.

表 1　计算效率比较

Table 1　Comparison of computation efficeency

算法类型	节点个数	连接个数	生成一个解的计算量	总计算量
动态窗口蚁群算法	$Nw+1$	$w+(N-1)w^2$	Nw	$O[C(TMNw+Nw^2)]$
一般蚁群算法	Nq^r+1	$q^r+(N-1)q^{2r}$	Nq^r	$O(CMNq^r)$

从表 1 的比较结果可以看出，动态搜索窗口的蚁群算法的计算量与一般蚁群算法计算量的比值 (设 C, N 等算法参数相等) 约等于

$$\frac{O[C(TMNw+Nw^2)]}{O(CMNq^r)} \approx \frac{C(TMNw+Nw^2)}{CMNq^r} = \frac{TMw+w^2}{Mq^r} \tag{13}$$

窗口宽度 w 可以根据具体问题来设置，一般当 r 较大时远小于 q^r，故这个比值远小于 1. 因此，尽管动态窗口蚁群搜索采用遗传优化以产生搜索窗口，但总体计算量还是小的.

4　仿真实验

考虑以下多输入、连续、非线性、时变动态系统的最优控制问题

$$J(N)=\sum_{t=0}^{N-1}\left\{\left|\cos\left[\frac{\pi}{2}x_1(t)x_2(t)\right]\right|+\sqrt{u_1^2(t)+u_2^2(t)}\right\} \tag{14}$$

$$x_1(t+1) = (t+1)\sin\left[\frac{\pi}{2}x_1(t)\right] + x_2(t)u_1(t) \tag{15}$$

$$x_2(t+1) = (t+1)x_1(t)\cos\left[\frac{\pi}{2}x_2(t)\right] + u_2(t) \tag{16}$$

$$|u_1(t)| \leqslant 1; \quad |u_2(t)| \leqslant 1; \quad t = 0, 1, \cdots, N-1 \tag{17}$$

设 $N = 10$, $x_1(0) = 0$, $x_2(0) = 0$, 分别用一般蚁群算法和动态窗口蚁群优化算法对问题进行求解. 在一般蚁群算法中将决策变量均匀离散化, 并令 u_1 与 u_2 的离散段数皆为 $q = 200$(即精确到 0.01, 每层有 200^2 个节点). 在动态窗口蚁群优化算法中设搜索窗口宽度为 $w = 200$, 每隔 $T = 3$ 次蚁群迭代进行一次遗传操作. 用 20 个蚂蚁进行迭代, 其他参数完全一致. 为方便比较, 对两种算法各进行 100 次仿真计算, 并且每次各迭代 150 次, 其结果如表 2 所示.

表 2 100 仿真结果的比较

Table 2 Comparison of simulation results for 100 times

算法类型	最好解的目标函数值 1)	解的平均目标函数值
动态窗口蚁群算法	5.408112	6.31608
一般蚁群算法	5.784397	6.665496

1) 100 次仿真结果中最好的解

为比较两种算法的一次仿真计算过程, 再让其各迭代 450 次, 其搜索过程中所得到的最优解曲线如图 2 所示.

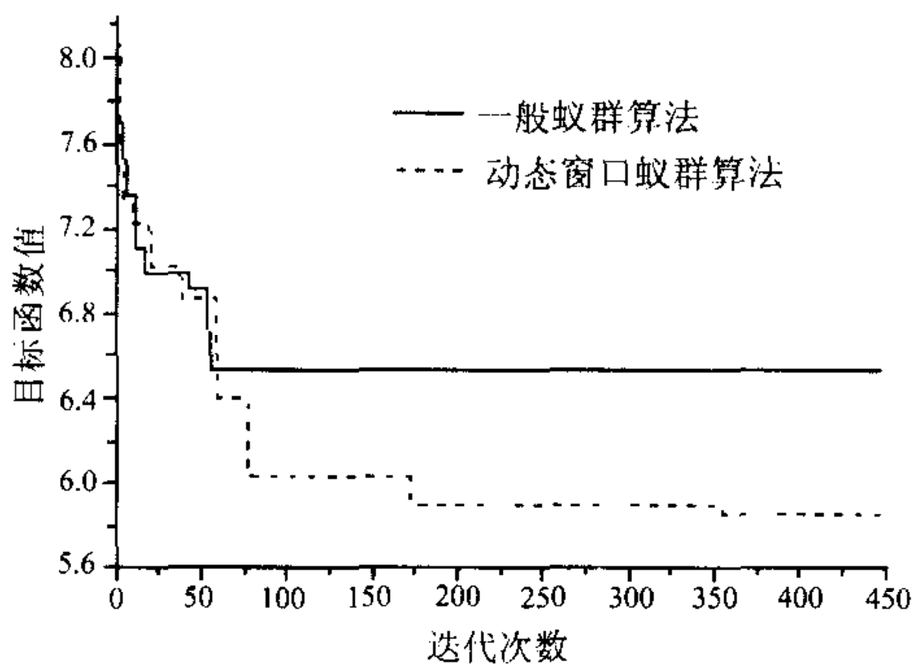

图 2 一次搜索进程中的最优解曲线

Fig. 2 curve of optimal result in one search process

将算法参数代入公式 (13), 得到的该问题的动态窗口蚁群算法与一般蚁群算法的平均计算量的比值为

$$\frac{3 \times 20 \times 200 + 200^2}{20 \times 200^2} \approx \frac{1}{15}$$

对表 2 、图 2 中两种蚁群算法的仿真结果和迭代过程的比较可以发现, 动态窗口蚁群优化算法能比一般蚁群算法搜索到更好的解. 其原因已在 3.1 节中做了详细的分析, 是对搜索窗口的动态调整和对上游连接信息素的重新分配有效地避免了蚁群算法的 “停滞 ” 现象, 从而使搜索范围更加广泛而有效. 例如, 图 2 上的一般蚁群算法搜索过程的最优解曲线的 “停滞 ” 现象就比动态窗口蚁群算法明显的多.

5　结语

经过对算法原理和计算效率的分析, 以及仿真实例的研究, 我们发现具有动态搜索窗口的蚁群算法与一般蚁群算法和遗传算法相比有以下一些优点:

1) 蚂蚁在各阶段排斥产生非法解的决策值, 解决了 GA 中描述复杂约束条件的问题;

2) 通过动态调整各阶段的搜索窗口和对信息素的重新分配, 有效地避免了蚁群算法的 "停滞 " 问题, 从而搜索到了更好的解;

3) 大大降低了一般蚁群算法求解复杂多阶段决策问题的内存使用量和计算量;

4) 所求解问题的范围也大大增加, 方程 (1) ~ (7) 归纳了该算法所能求解的多阶段决策问题的抽象数学形式. 对于离散问题, 可用整数编码或二进制编码的遗传算子更新各阶段决策域的搜索窗口, 例如在其应用于大范围交通协调控制问题的研究中, 也取得较好的性能.

References

1 Hu Yun-Quan, Guo Yao-Huang. Tutorial of Operational Research. Peking: Tsinghua University Press, 1998 (in Chinese)

2 Wann-Ming Wey. Model formulation and solution algorithm of traffic signal control in an urban network. *Computers, Environment and Urban Systems*, 2000, **24**(4): 355 ~ 377

3 Hong K Lo. Dynamic network traffic control. *Transportation Research Part A*, 2001, **35**(8): 721 ~ 744

4 Xie Xue-Shu. Theory and Application of Optimal Control. Peking: Tsinghua University Press, 1986 (in Chinese)

5 Chen Guo-Liang, Wang Xufa, Zhuang Zhenquan, Wang Dongshen. Theory and Application of Genetic Algorithm. Peking: Posts & Telecom Press, 1996 (in Chinese)

6 M Dorigo, L M Gambardela. The ant system: optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man & Cybern.*, 1996, **B26**(1): 29 ~ 41

7 M Dorigo, E Bonabeau, G Theraulaz. Ant algorithms and stigmergy. *Future Generation Computer Systems*, 2000, **16**(8). 851 ~ 871

闻　育　博士研究生. 研究方向为交通仿真和控制、智能计算方法.

(**WEN Yu**　Ph.D. Candidate. Research fields consist of intelligent computation, traffic simulation and control.)

吴铁军　浙江大学智能系统与决策研究所所长、教授、博士生导师. 研究领域为智能系统控制与优化, 用于最优化的计算智能, 机器学习及其在复杂系统中的应用等.

(**WU Tie-Jun**　Professor, Ph.D. director and director of institute of intelligent system and decision making, Zhejiang university. Research fields consist of system intelligent control and optimization, intelligence in optimization computation, theory and application of machine learning in complex system.)