

基于自适应蚁群算法的 QoS 组播路由算法

陈杰¹, 张洪伟²

(四川大学计算机学院, 成都 610064)

摘要: 提出一种改进的自适应蚁群优化算法, 在信息素更新策略中引入全局最优系数, 研究多约束条件下的 QoS 组播路由问题。动态更新信息素能够确保自适应地改进全局搜索能力和收敛性能, 避免陷入局部最优解。仿真结果表明, 该算法比蚂蚁-遗传算法在解决多约束条件下的 QoS 组播路由问题时更有效。

关键词: 蚁群算法; 服务质量; 组播路由; 自适应

QoS Multicast Routing Algorithm Based on Adaptive Ant Colony Algorithm

CHEN Jie¹, ZHANG Hong-wei²

(College of Computer Science, Sichuan University, Chengdu 610064)

【Abstract】 This paper proposes an improved adaptive colony optimization algorithm and mainly presents a parameter of global best tour in strategies of pheromone updating to discuss QoS multicast routing problem with multiple constrains. The dynamic pheromone updating is adopted to ensure that global searching and convergence abilities are improved adaptively and avoid falling in local peak. Simulation results demonstrate that the proposed algorithm performs better than the combination of ant colony algorithm and genetic algorithm for solving QoS multicast routing problem with multiple constrains.

【Key words】 ant colony algorithm; QoS; multicast routing; adaptive

1 概述

随着 Internet 网络的广泛应用和规模的持续扩大, 与多媒体相关的实时业务也在逐步增长。由于多媒体实时业务对网络服务质量(QoS)提出了越来越高的要求, 使得 QoS 网络路由成为网络技术领域的研究热点之一。QoS 网络路由的目的是在网络中寻找一条最佳路径, 从源节点出发, 经历所有的目的节点, 并满足网络路由中的延时、延时抖动、带宽、丢包率、代价等约束条件以保证网络的服务质量。

20 世纪 90 年代, 意大利学者 M. Dorigo, V. Maniezzo 等人通过模拟自然界蚂蚁寻径的行为, 提出了一种全新的启发式算法——蚁群算法(Ant Colony Algorithm, ACA), 它被广泛用于解决各种具有 NP 难的问题^[1]。Wang Z 等学者已经证明了包含 2 个及以上的约束条件的 QoS 网络路由是一个 NP 完全问题^[2]。

该文提出一种基于改进的自适应蚁群优化算法, 并应用在多约束条件下的 QoS 网络路由。

2 基本蚁群系统模型

蚁群算法是一种基于自然界中蚂蚁种群行为的新优化算法。在实验观察中, 蚂蚁在寻找食物的过程中, 会在所经过的路径上留下信息素。由于蚂蚁具有倾向于选择信息素强度高的路径, 则后来的蚂蚁选择该路径的概率也就越大。同时, 路径上的信息素也会随着时间的流逝而不断地挥发, 这种机制使得蚂蚁不完全受过去经验的约束, 有利于蚂蚁向新的路径搜索。蚁群在寻找食物源到蚁巢的最优路径是蚁群相互协作的结果, 信息素强度最高的路径即为最优路径。

蚁群系统^[1]是由 Dorigo 和 Gmabardella 在 1996 年提出

的, 有效地改进了蚂蚁系统的性能。蚁群系统对蚂蚁系统主要做了 3 个方面的改进:

(1) 为了避免停滞现象和搜索新的路径, 使用了状态转移规则, 它采用确定性选择和随机性选择相结合的选择策略。状态转移规则给出了一只位于节点 r 的蚂蚁 k 选择的下一个城市 s 。规则如下式所示:

$$s_k = \begin{cases} \arg \max_{u \in allowed_k} \{[\tau(r,u)]^\alpha [\eta(r,u)]^\beta\}, & q < q_0 \\ S, & q > q_0 \end{cases} \quad (1)$$

$$P_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t) \eta_{ij}^\beta(t)}{\sum_{s \in allowed_k} \tau_{ij}^\alpha(t) \eta_{ij}^\beta(t)}, & j \in allowed_k \\ 0, & j \notin allowed_k \end{cases} \quad (2)$$

其中, s_k 表示蚂蚁 k 所选择的下一个节点; q 是在 $[0, 1]$ 之间的一个随机数; q_0 为参数 ($0 < q_0 < 1$)。式(1)中的 S 根据式(2)计算出下一个访问节点; $allowed_k$ 表示蚂蚁 k 下一步允许选择的节点的集合; α 表示信息启发因子, 表示轨迹的相对重要性; β 为期望启发式因子, 表示能见度的相对重要性; τ_{ij} 表示节点 i 和节点 j 之间的信息素强度; η_{ij} 表示由节点 i 转移到节点 j 的启发程度。在该文仿真中, $\eta_{ij} = \frac{1}{cost(i,j)}$, $cost(i,j)$ 表示节点 i, j 之间的费用大小。

作者简介: 陈杰(1982-), 男, 硕士研究生, 主研方向: 计算机网络与数据库; 张洪伟, 教授、博士后

收稿日期: 2007-08-20 **E-mail:** chenjie2010@sina.com

(2)使用了全局更新规则,即在每次循环过程结束,只允许全局最优的蚂蚁释放信息素。这样使得蚂蚁更倾向于选择最优路径,缩短了收敛于最优解的时间。全局更新规则如下式所示:

$$\tau(r,s) \leftarrow (1-\alpha)\tau(r,s) + \alpha \Delta\tau(r,s) \quad (3)$$

$$\Delta\tau(r,s) = \begin{cases} Q/L_{\text{best}} & \text{if } (r,s) \in \text{全局最优路径} \\ 0 & \text{否则} \end{cases} \quad (4)$$

其中,式(3)中的 $\Delta\tau(r,s)$ 通过式(4)来计算。 α 表示全局信息素挥发参数, $0 < \alpha < 1$; L_{best} 为所有蚂蚁完成一次寻径后的全局最优路径; Q 为常数,表示节点之间的初始化的信息素强度。

(3)使用了局部更新规则,降低蚂蚁经过路径上的信息素的强度,避免蚂蚁过早地收敛于同一解。局部更新规则如下:

对于第 k 只蚂蚁,如果节点 r, S 是它所选择路径上的 2 个相邻节点,则有

$$\tau(r,s) \leftarrow (1-\rho) \times \tau(r,s) + \rho \times \tau_0 \quad (5)$$

$$\tau_0 = (nL_{\text{min}})^{-1} \quad (6)$$

其中,式(5)中的 τ_0 通过式(6)来计算; ρ 表示局部信息素挥发参数; n 表示节点数量; L_{min} 是由启发式邻近算法得到最短的路径长度。

3 QoS 网络路由模型的自适应蚁群算法

3.1 算法的改进模型

蚁群算法作为一种新型的进化算法,其正反馈机制有利于提高发现较优解的能力,分布式计算能够避免早熟收敛和陷入局部最优解,贪婪启发式算法有助于在搜索过程的早期阶段找到满足条件的解,减少了搜索时间。但是该算法也存在一些缺点:(1)搜索时间长;(2)易于陷入局部最优解。在搜索过程中,尽力加快收敛速度又尽量避免陷入局部最优解,是值得研究的问题。

该算法的改进在于首先引进全局最优系数: $\delta = n/N$, n 表示最优解在连续 n 次循环过程后均没有发生变化; N 表示初始化时设置的循环总次数。然后根据 δ 作为条件来自适应改变 ρ, Q 的值。

(1)自适应改变 ρ 的值。如果 ρ 的值过大,由于全局更新规则只对全局最优路径更新,因此算法容易陷入局部最优解的状态。如果 ρ 的值过小,虽然提高了算法的全局搜索能力,但是降低了收敛速度。所以,可以自适应地改变 ρ 的值,以提高收敛速度的同时不陷入局部最优解。

在连续 n 次循环过程最优解都没有变化,即当 $\delta(t_n) > \delta$ 时,说明搜索过程可能陷入了局部最优解,此时,根据每条路径上的信息素强度来实时调整该路径上的信息素挥发参数 $\rho_{ij}(t)$ 的值。 $\rho_{ij}(t)$ 的值与蚁群算法的不同之处是在陷入局部最优解后自适应地改变 ρ 的值;与文献[3]的不同之处是在 t 时刻路径的信息挥发参数不仅与时间相关,还与其路径的信息素强度相关。 ρ 的调整公式如下:

$$\rho_{ij}(t) = \begin{cases} \rho_{ij}(t-1) \times \frac{\tau_{ij}(t-1)}{\tau_{ij}^*(t)} & \text{当 } \delta(t) > \delta \text{ 时, 且 if } (i,j) \in \text{最优路径} \\ \rho_{ij}(t-1) & \text{否则} \end{cases} \quad (7)$$

其中, $\rho_{ij}(t)$ 表示在 t 时刻,路径 (i,j) 上的信息挥发参数,取值范围在区间 $[\rho_{\text{min}}, 1)$ 内, ρ_{min} 为 ρ 的最小值; $\tau_{ij}^*(t)$ 是在全局更新规则中,在 $\rho_{ij}(t) = \rho_{ij}(t-1)$ 这个条件下求出来的一个参数值。

(2)自适应改变 Q 的值。最优解在连续 n 次循环过程有变化,即当 $\delta(t_n) < \delta$ 时,说明搜索过程仍在全局范围进行,未陷入局部最优。此时可以在全局更新规则中动态调整信息素的方法,采用时变函数 $Q(t)$ 来代替为常数项的信息素强度 Q 。与文献[4]不同的是,文献[4]自适应改变 Q 的值为了解决出现的停滞现象和陷入局部最优等问题。

$$Q(t) = \begin{cases} \gamma Q(t-1) & \text{当 } \delta(t_n) < \delta \text{ 时} \\ Q(t-1) & \text{当 } (\delta(t_n) \geq \delta) \text{ 时} \end{cases} \quad (8)$$

由于正反馈机制和信息挥发机制的存在,因此针对路径上的信息素强度可能过小或是过大的特点,需要对路径信息素强度做最大和最小限制。即 $\forall \tau_{ij}(t)$, 有

$$\min_{t \rightarrow \infty} \tau_{ij}(t) \leq \tau_{ij}(t) \leq \max_{t \rightarrow \infty} \tau_{ij}(t)$$

根据最大-最小蚂蚁系统(Max-Min Ant System, MMAS)^[5] 的研究,将信息素的初始值设为 τ_{max} , 每条边上的信息素的值的范围限制在 $[\tau_{\text{min}}, \tau_{\text{max}}]$ 区间内。

3.2 QoS 网络路由模型

设 $N < V, E >$ 表示网络,其中, V 表示为网络中节点的集合; E 表网络中双向链路的集合; $s \in V$ 为源节点; $M \in \{V - \{s\}\}$ 为目的节点。

对于任一网络节点 $n \in V$, 定义 4 种属性,分别为:延迟函数 $delay(n)$, 延迟抖动函数 $delay_jitter(n)$, 包丢失率函数 $packet_loss(n)$ 和费用函数 $cost(n)$ 。

对于任意链路 $e \in E$, 定义 4 种属性:延时函数 $delay(e)$, 延时抖动函数 $delay_jitter(e)$, 带宽函数 $bandwidth(e)$ 和费用函数 $cost(e)$ 。

对于给定的源点 $s \in V$, 终点集合 M 和源点 s 组成的组播树 $T(s, M)$ 存在下列关系:

$$delay(T(s, M)) = \sum_{e \in T(s, M)} delay(e) + \sum_{n \in T(s, M)} delay(n)$$

$$delay_jitter(T(s, M)) = \sum_{e \in T(s, M)} delay_jitter(e) + \sum_{n \in T(s, M)} delay_jitter(n)$$

$$bandwidth(T(s, M)) = \min\{bandwidth(e), e \in T(s, M)\}$$

$$cost(T(s, M)) = \sum_{e \in T(s, M)} cost(e) + \sum_{n \in T(s, M)} cost(n)$$

$$packet_loss(T(s, M)) = 1 - \prod_{n \in T(s, M)} (1 - packet_loss(n))$$

QoS 组播路由问题就是在网络 $N(V, E)$ 中,给定源节点 s , 终点集合 M , 寻找一棵组播树 $T(s, M)$ 满足以下约束条件:

- (1)延迟约束: $delay(T(s, M)) \leq D$;
- (2)带宽约束: $bandwidth(T(s, M)) \geq B$;
- (3)延迟抖动约束: $delay_jitter(T(s, M)) \leq DJ$;
- (4)包丢失率约束: $packet_loss(T(s, M)) \leq PL$;
- (5)费用约束,在满足前面 4 个约束的条件下, $cost(T(s, M))$ 最小。

其中, B, D, DJ 和 PL 分别为网络中的带宽、延迟、延迟抖动和包丢失率等的约束限制。

目标函数值:

$$L_k = \frac{\lambda}{cost[T(s, M)]} (Af_d + Bf_{dj} + Cf_{pl}) \quad (9)$$

$$f_d = \Phi_d(delay(P_T(s, u)) - D); (\Phi_d(z) = \begin{cases} 1, z \leq 0 \\ r_d, z > 0 \end{cases};)$$

$$f_{dj} = \Phi_{dj}(delay_jitter(P_T(s, u)) - DJ); (\Phi_{dj}(z) = \begin{cases} 1, z \leq 0 \\ r_{dj}, z > 0 \end{cases};)$$

$$f_{pl} = \Phi_{pl}(delay(P_T(s, u)) - PL); (\Phi_{pl}(z) = \begin{cases} 1, z \leq 0 \\ r_{pl}, z > 0 \end{cases};)$$

其中, λ 是正实系数; A, B, C 分别为 f_d, f_{dj}, f_{pl} 的正加权系数, 分别表示延时、延时抖动和丢包率在目标函数值中所占的比重; $\Phi_d(z)$ 是延时度量的惩罚函数, 当满足延时约束 ($delay(P_T(s, u)) \leq D$) 时, 其值为 1, 否则等于 r_d ($0 < r_d < 1$); $\Phi_{dj}(z)$ 是延时抖动度量的惩罚函数, 当满足延时抖动约束 ($delay_jitter(P_T(s, u)) \leq DJ$) 时, 其值为 1, 否则等于 r_{dj} ($0 < r_{dj} < 1$); $\Phi_{pl}(z)$ 是丢包率度量的惩罚函数, 当满足丢包率约束 ($delay(P_T(s, u)) \leq PL$) 时, 其值为 1, 否则等于 r_{pl} ($0 < r_{pl} < 1$); r_d, r_{dj}, r_{pl} 三者的值的大小决定惩罚的程度^[6]。在本文仿真中, $\lambda=1, A=B=C=1, r_d=r_{dj}=r_{pl}=0.5$ 。

改进后的算法求解问题的步骤如下:

(1)初始化参数。假定网络中有 m 只蚂蚁, n 个节点, 循环次数为 N 。确定 $\alpha, \beta, q_0, \delta$ 等的值, 给出各个节点(d, dj, pl, c)的值, 以及每条存在边(d, dj, b, c)的值和给出约束条件的 D, DJ, B, PL 的值。每条边(i, j)的信息素浓度设置初始值为 $Q(t_0)$, 挥发系数为 $\rho(t_0)$ 。NC 为当前循环次数, 初始值设置为 0。

(2)精简网络。根据约束条件(2), 精简网络拓扑结构中不符合要求的链路或节点。

(3)初始化禁忌表 $tabu_k$ 。设蚂蚁 k 形成的组播树为 T_k , 链路集合为 E_k, V 表示网络中节点的集合。初始化时, 将组播源点 s 加 $tabu_k$ 表中, E_k 为空。每只蚂蚁独立地从 $tabu_k$ 中的节点开始, 根据状态转移规则式(1), 从 $V-tabu_k$ 中选择一个节点下一步所走的相对最优路径(i, j), 并判断从组播源点 s 到该点所经过的路径和节点是否满足约束条件: 1)延时; 2)延时抖动; 3)丢包率。如果满足以上 3 个条件, 将相对最优路径(i, j)加入到 E_k 中, 否则继续从 $V-tabu_k$ 中选择其他满足条件的节点。选择的节点满足条件后, 使用局部更新规则式(5)更新相应路径上的信息素, 并将节点加入到禁忌表 $tabu_k$ 。直至所有蚂蚁都完成一次路径搜索。

(4)对蚂蚁寻找组播树进行组播树枝剪。枝剪规则为: 如果组播树的每条路径的叶子节点不是目标节点, 则从 $tabu_k$ 中删除该节点, 并从 E_k 中删除与该节点相连的链路。

(5)通过式(6), 对组播树集合 T 中各个组播树的路径集合 E_k 和顶点集合 $tabu_k$, 计算对应的目标函数值 L_k , 并进行比较, 获得当前迭代最优组播树和全局最优组播树。

(6)计算全局最优系数 δ , 并根据式(7)和式(8)计算 $\rho_{ij}(t)$ 和 $Q(t)$, 然后由全局更新规则式(3), 将全局最优组播树中路段的信息素增量并更新。

(7)若 $NC=N$, 转步骤(8); 否则清空禁忌表 $tabu, NC=NC+1$, 再转步骤(3)。

(8)输出满足约束条件的最优目标函数值及所对应的组播树(即为最优链路); 若整个蚁群的所有路径都不满足约束条件, 则所求链路不存在。

4 仿真实验及其比较

通过程序实现了该文提出的自适应蚁群算法在解决网络模型的应用, 使用 C# 语言在 .NET Framework 2.0 环境下实现。

为了方便与文献[7]做比较, 这里选择的网络模型与文献[7]一致, 如图 1 所示。在各个节点的(d, dj, pl, c)和每条存在边的(d, dj, b, c)的描述中, d, dj, pl, b, c 分别表示延时、延

时抖动、丢包率、带宽、代价。

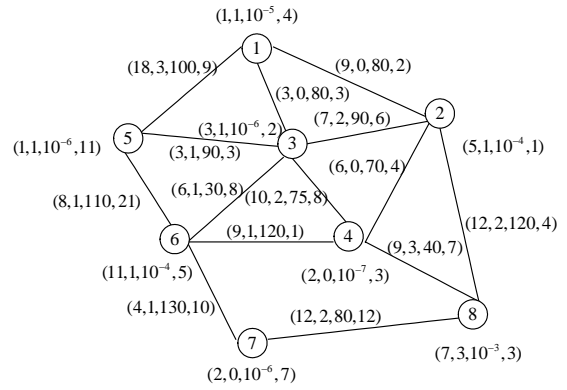


图 1 8 个网络节点网络结构模型

仿真实验以节点 1 为源节点, 以节点 2、节点 4、节点 5、节点 7 为目的节点, 寻找满足约束条件的最优路径。该网络模型的约束条件为 $B=70, D=46, DJ=8, PL=0.01$ 。

仿真实验的步骤如下:

(1)选择 QoS 网络模型, 如图 1 所示。

(2)初始化约束条件, 根据网络的带宽条件, 去掉图 1 中节点(3, 6)之间的边和(4, 8)之间的边。

(3)选择蚂蚁的数目为 5, 迭代寻找的次数 $N=20$ 。并设置 $\alpha, \beta, q_0, \delta, \gamma, \rho(t_0), Q(t_0)$ 等的值, 采用 MMAS 算法对 τ_{ij} 进行限制。

(4)使用该文提出的算法, 5 只蚂蚁从源节点 1 出发, 经历目的节点 2、节点 4、节点 5、节点 7, 寻找满足约束条件的最优组播树。

(5)在一次寻找结束后, 重复步骤(3)和步骤(4), 设置不同的参数, 进行多次的仿真实验。通过大量仿真实验, 获得选择的参数 $\alpha=1, \beta=1, q_0=0.1, \delta=0.2, \gamma=1.05, \rho(t_0)=0.1, Q(t_0)=10$ 。

根据上述步骤, 在这组参数下从仿真实验中获得最优组播路由结果: (1)最优组播树: (1,2), (1,3), (3,5), (2,4), (4,6), (6,7)。(2)最大延时: 46。(3)最大延时抖动: 4。(4)最大丢包率: 0.000 2。(5)最小总代价 40。最优组播树路径如图 2 所示。延迟、延迟抖动、带宽、丢包率和最小代价等多约束条件下求解的延迟、延迟抖动、代价曲线如图 3、图 4 所示。

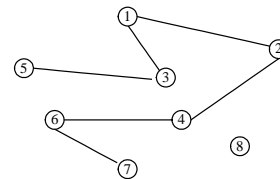


图 2 结果图

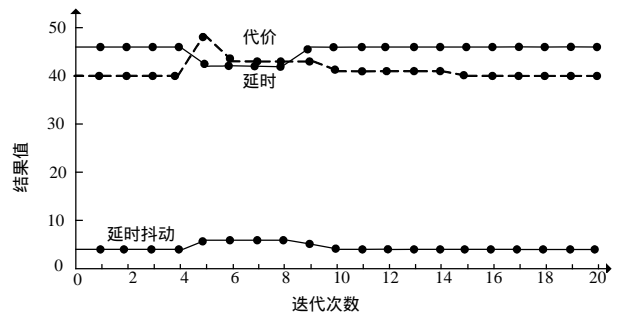


图 3 本文算法的曲线

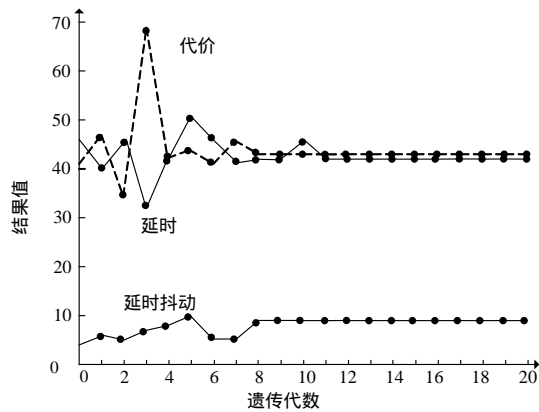


图4 文献[7]的曲线

在组播树中，延迟抖动为经过该组播树各条路径延迟抖动中的最大值，延迟也为经过该组播树各条路径延迟中的最大值。本文算法引进全局最优系数 δ 后，曲线图中的代价曲线第1次出现的波动部分说明扩大全局的搜索能力，后趋于平缓说明加快了收敛速度。从图3与图4的对比中看出，本文算法的延迟抖动曲线、延迟曲线以及代价曲线比蚂蚁—遗传算法的曲线均要平稳，能够更快地找到最优解，而且在满足多约束条件下的最优组播树的代价优于文献[7]的结果。

5 结束语

该文提出了一种自适应蚁群优化算法，通过在仿真实验中求解 QoS 网络路由包含延迟、延迟抖动、带宽、丢包率和

最小代价等约束条件在内的 QoS 组播路由问题，有效地加快收敛速度和避免陷入局部最优解，提高了算法寻找最优解的效率。

参考文献

- [1] Dortgo M, Maniezzo V, Colomi A. Ant System: Optimization by a Colony Cooperating Agents[J]. IEEE Transactions on Systems, Man, and Cybernetics-bart B: Cybernetics, 1996, 26(1): 2941.
- [2] Wang Zheng, Crowcroft J. Quality of Service Routing for Supporting Multimedia Applications[J]. IEEE Journal on Selected Areas in Communications, 1996, 14(7): 1228-1234.
- [3] 王颖, 谢剑英. 一种自适应蚁群算法及其仿真研究[J]. 系统仿真学报, 2002, 14(1): 31-33.
- [4] 覃刚力, 杨家本. 自适应调整信息素的蚁群算法[J]. 信息与控制, 2002, 31(3): 1-33.
- [5] Stutzl T, Hoos H. Improvements on the Ant System: Introducing MAX-MIN Ant System[C]//Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms. [S. l.]: Springer-Verlag 1997: 245-249.
- [6] 王征应, 石冰心. 基于启发式遗传算法的 QoS 组播路由问题求解[J]. 计算机学报, 2001, 24(1): 55-61.
- [7] 孙力娟, 王汝传. 基于蚁群算法和遗传算法融合的 QoS 组播路由问题求解[J]. 电子学报, 2006, 34(8): 1391-1395.

(上接第 199 页)

背景。因此衡量尺度定义为：生成的背景和对应标准背景差图像的均值。

表1 不同背景生成算法在不同情况下的性能比较

方法	目标在场景中实现 (以均方差为尺度)	目标停留片刻 (以均方差为尺度)	目标离开、光照突变 (以均值为尺度)
Stauffer 等的多高斯分布背景生成法	2.97	18.86	4.46
本文的多高斯分布背景生成算法	2.05	6.53	4.01

通过比较可知，本文的多高斯分布背景生成算法在几种情况下都能生成出接近标准背景的背景图像，在运动目标短暂出现、运动目标停留、光照变化等方面，其性能都优于传统的多高斯分布法。

4 结束语

本文介绍了一个改进的多高斯分布背景生成算法。该算法与传统算法相比，不仅能顺利处理常见的情况，而且对光照的变化、场景中物体的来回摆动(如风中的树木、飘动的旗帜等)、目标慢速运动等都能有效地处理。实验结果证明，本文算法具有更好的适应能力和稳定性。

参考文献

- [1] Koller D, Weber J, Malik J. Robust Multiple Car Tracking with Occlusion Reasoning[C]//Proc. of the 3rd European Conference on Computer Vision. Stockholm, Sweden: Springer, 1994: 189-196.
- [2] Ridder C, Munkelt O, Kirchner H. Adaptive Background Estimation and Foreground Detection Using Kalman Filtering[C]//Proc. of ICAM'95. Istanbul, Turkey: [s. n.], 1995: 193-199.
- [3] Rittscher J, Kato J, Joga S, et al. Probabilistic Background Model for Tracking[C]//Proc. of the 6th European Conference on Computer Vision. Dublin, Ireland: [s. n.], 2000: 336-350.
- [4] Stenger B, Ramesh V, Paragios N, et al. Topology-free Hidden Markov Models: Application to Background Modeling[Z]. (2001-10-30). <http://www.citeseer.ist.psu.edu/stenger01topology.html>.
- [5] Wren C R, Azarbayejani A, Darrell T, et al. Finder: Real-time Tracking of the Human Body[J]. IEEE Trans. on Pattern Analysis and Machine Intelligence, 1997, 19(7): 780-785.
- [6] Stauffer C, Grimson W E L. Adaptive Background Mixture Models for Real-time Tracking[J]. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Fort Collins, Colo, USA: [s. n.], 1999: 2246-2252.