

文章编号: 1001-0920(2009)03-0383-05

基于协同进化的平面追逃对策研究

陈迎春¹, 齐欢²

(1. 中国人民解放军 91388 部队, 广东 湛江 524022; 2. 华中科技大学 系统工程研究所, 武汉 430074)

摘要: 建立了二维平面内动力学约束下追逃运动的数学模型. 首先为追捕者设计了基于比例制导算法和进化算法的混合追捕策略, 以提高其追捕能力; 然后利用协同进化算法对追捕者和逃跑者的追逃策略进行进化. 仿真结果表明, 进化后的逃跑策略能有效规避比例制导的追捕者, 逃跑者在协同进化过程中涌现出众多复杂多变的规避策略.

关键词: 协同进化; 微分对策; 追逃对策; 博弈论

中图分类号: TP391.9

文献标识码: A

Co-evolutionary pursuit-evasion game on a plane

CHEN Ying-chun¹, QI Huan²

(1. Unit 91388 of PLA, Zhanjiang 524022, China; 2. Institute of Systems Engineering, Huazhong University of Science and Technology, Wuhan 430074, China. Correspondent: CHEN Ying-chun, E-mail: chyc@21cn.com)

Abstract: A two-dimensional mathematical model is established for the movement of pursuers and evaders with dynamical constraints. Based on proportional guidance principle, a mixed strategy is designed for the pursuers to guarantee the co-evolution process between the pursuers and evaders. Co-evolutionary algorithm is applied to evolve the strategies of both players simultaneously. The evolved evading strategy can effectively avoid proportionally guided pursuers. During the co-evolutionary process, many complex strategies emerge.

Key words: Co-evolutionary algorithm; Differential game; Pursuit-evasion game; Game theory

1 引言

在经典的 Homicidal chauffeur 追逃微分对策^[1]中, 汽车 w 和机器人 m 分别以恒定速率 v_w 和 v_m 在平面上运动 ($|v_w| > |v_m|$), 汽车的转弯半径为 R_w , 机器人的转弯半径 $R_m = 0$, 即机器人运动较慢但机动能力较强. 逃逸机动对策^[2]假定 w 能根据 m 当前的运动状态预测其运动轨迹, 并能在最短时间内捕获到 m 的位置运动; w 改变方向时速度会降低, 直线运动时可以加速. 这两种对策的解是高度非线性的, 而逃逸机动的对策解更为复杂, 其中两个追捕者的逃逸机动对策尚未发现最优策略, 因此难以通过数学分析方法求解. 逃逸机动对策可用于描述军事领域的许多问题 (如飞机规避导弹), 因此对逃逸机动对策的研究有着重要的现实意义.

Hillis^[3] 首次将捕食者与猎物之间的协同进化作为计算模型. Miller 和 Cliff^[4] 对协同进化追逃问题进行综述, 奠定了协同进化追逃的生物学和博弈

论的基础. Reynolds^[5] 在研究贴标签游戏中探讨了追逃战术的协同进化问题. 王泉德^[2] 假定逃跑者和追捕者具有零转弯半径, 这与现实情况不符. 另外, 追捕者采取的固定策略也容易被逃跑者发现并采取反制措施. 为此, 本文用协同进化的思想研究二维平面上两个追捕者和一个逃跑者之间的追逃对抗问题.

2 协同进化的理论基础

现实世界的许多问题都能以极小极大问题来建模, 例如导弹与目标之间的对抗. 为了求解约束条件下的极小极大问题, Tahk 和 Sun^[6] 提出用协同进化增广拉格朗日法^[6-8], 将两个利益冲突局中人的零和博弈近似为静态矩阵对策后再求鞍点解. 该方法的主要优点是不需要梯度信息和初始估计, 因此非常适用于具有复杂动力学约束的追逃问题. 在该方法中, 参数向量组的策略是极小化增广拉格朗日函数, 而拉格朗日乘子组的目标是极大化, 每个组都

收稿日期: 2008-01-24; 修回日期: 2008-06-23.

基金项目: 国家自然科学基金项目 (60774036); 航天支撑基金项目 (034. 1JW05).

作者简介: 陈迎春 (1972 →), 男, 湖南湘乡人, 高级工程师, 博士, 从事计算机仿真、系统建模与分析的研究; 齐欢 (1948 →), 男, 武汉人, 教授, 博士, 从事复杂系统分析与集成等研究.

独立进化以产生最佳收益的个体^[7]. 有界空间 S 内约束优化问题可描述为

$$\begin{aligned} \min_x f(x), x \in S \subset R^n; \\ \text{s.t. } g_i(x) = 0, i = 1, 2, \dots, m, \\ h_j(x) = 0, j = 1, 2, \dots, l. \end{aligned} \quad (1)$$

其增广拉格朗日对偶问题可表示为

$$\begin{aligned} \max_{\lambda, \mu} L_A(\lambda, \mu, x) = \max_x \min_x L_A(x, \lambda, \mu), \\ R^k, k = m + l; \\ \text{s.t. } \lambda_i = 0, i = 1, 2, \dots, m. \end{aligned} \quad (2)$$

其中增广拉格朗日函数 L_A 可表示为

$$L_A(x, \lambda, \mu) = f(x) + \sum_{i=1}^m \lambda_i (g_i(x) + \frac{1}{2} h_i^2(x)) + \sum_{i=1}^l \mu_{m+i} h_i^2(x). \quad (3)$$

第 i 个不等式约束的处罚项 λ_i 为

$$\lambda_i(x, \lambda_i) = \begin{cases} \lambda_i (g_i(x) + \frac{1}{2} h_i^2(x)), & g_i(x) \geq -\lambda_i / 2; \\ -\lambda_i^2 / 4, & g_i(x) < -\lambda_i / 2. \end{cases} \quad (4)$$

式(2)表示对偶问题是极大极小问题,求解矩阵对策便可得到原问题的解. 假设在零和博弈 G 中, $u \in U \subset R^n$ 的目标是极小化收益函数 $F = F(u, v)$, $v \in V \subset R^m$ 的目标是极大化收益函数, G 可近似为静态的矩阵对策 G_M .

设 i 和 j 分别表示种群 X 和 Y 中的个体,其策略分别为 u^i 和 v^j . 将 i 和 j 之间的对策值用 $F(u^i, v^j)$ 表示,则三元组 (F, X, Y) 便确定了矩阵对策 G_M ,局中人 u 和 v 具有如下有限策略:

$$\begin{cases} u^i \in U_M \subset U \subset R^n, i = 1, 2, \dots, N_u; \\ v^j \in V_m \subset V \subset R^m, j = 1, 2, \dots, N_v. \end{cases} \quad (5)$$

其中 N_u 和 N_v 分别为 u 和 v 的策略个数. 设 u^s 为 u 的安全策略, $\bar{F}(G_M)$ 为 u 损失的安全水平;同理, v^s 为 v 的安全策略, $E(G_M)$ 为 v 收益的安全水平. 则局中人 u 和 v 在安全策略下的收益分别满足不等式

$$\begin{aligned} \bar{F}(G_M) &= \max_j F(u^s, v_j) = \max_j F(u_i, v_j), \\ E(G_M) &= \min_i F(u_i, v^s) = \min_i f(u_i, v_j), \\ i &= 1, 2, \dots, N_u, j = 1, 2, \dots, N_v. \end{aligned}$$

一般情况下,安全水平满足

$$E(G_M) = \bar{F}(G_M). \quad (6)$$

如果式(6)等号成立,则 (u^s, v^s) 即为 G_M 的鞍点解. 由于 G_M 只是 G 的近似,安全策略 (u^s, v^s) 可能与 G 的鞍点解 (u^*, v^*) 相去甚远,但是随着 U_M 和 V_M 在鞍点解附近种群密度的增加, (u^s, v^s) 会不断逼近 (u^*, v^*) .

3 二维平面追逃问题的模型描述

追逃模型考虑了追捕者 w 和逃跑者 m 的动力学约束. 追捕者的速度和转弯半径均大于逃跑者,即追捕者的速度较快但机动能力较弱. 追逃双方的策略根据与对手对抗的结果而不断进化. 追逃双方的符号定义和态势如图 1 所示^[9]. 其中: γ_w 为追捕者的视角角; v_w 和 v_m , R_w 和 R_m 分别为追逃双方的速度和航向角.

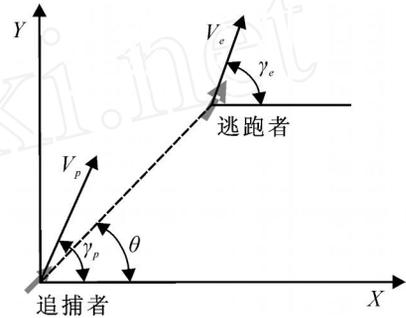


图 1 平面上追捕者和逃跑者的态势和符号定义
追逃双方的运动可描述如下:

$$\begin{cases} \dot{x}_i = v_i \cos \gamma_i; \\ \dot{y}_i = v_i \sin \gamma_i; \\ \dot{v}_i = \begin{cases} -\frac{v_i^2}{R_i} (a_i + b_i u_i^2), & |u_i| \leq 0.01; \\ c_i, & \text{其他} \end{cases} \\ \dot{\gamma}_i = \frac{v_i}{R_i} u_i. \end{cases} \quad (7)$$

其中:下标 $i = w, m$ 分别表示追捕者和逃跑者; (x_i, y_i) , v_i , R_i , a , b 和 u_i 分别表示位置、速度、最小转弯半径、阻力系数与牵引力系数比、牵引力系数和控制输入. $|u_i| \leq 1$, 若 $|u_i| \leq 0.01$, 则会因为改变运动方向而减速; 否则, 将以加速度 c_i 运动.

式(7)表示追逃双方的机动角度越大速度下降越快,且最大机动角度受最小转弯半径的约束,但机动角度较小时会加速. 给定式中的参数和初始参数,根据控制输入便可得到追逃态势. 若追捕者和逃跑者之间的距离小于追捕者的捕获半径,则追捕者获胜;若在规定的对抗时间内追捕者不能捕获逃跑者,则逃跑者获胜.

4 协同进化追逃算法的设计

4.1 追逃策略的设计

逃跑者的控制输入 u_m 是待优化参数,由进化过程给出. u_m 可离散化为

$$u_m = (u_{m1}, u_{m2}, \dots, u_{mn}), \quad (8)$$

其中 n 为离散化的步数. 假设对抗时间为 T 秒,则在 $[0, T/n]$ 内逃跑者的控制信号为 u_{m1} , 在 $(T/n, 2T/n]$ 内控制信号为 u_{m2} , 依此类推.

追捕者在追逃对抗中处于明显的劣势,初步仿真发现追捕者很难捕获逃跑者,因此为其增加了比例制导算法.采用比例制导的追捕者几乎总能捕获随机运动的逃跑者;逃跑者的策略如能进化,它将学会采取有效的机动策略摆脱追捕者.

追捕者的控制输入 u_w 由两部分组成,即

$$u_w = w_1 u_{w_1} + w_2 u_{w_2} \tag{9}$$

其中: $u_{w_1} = (u_{w_{1,1}}, u_{w_{1,2}}, \dots, u_{w_{1,N}})$ 为离散化的进化参数; u_{w_2} 为根据追逃态势计算得到的制导参数; w_1 和 w_2 分别表示进化参数和比例制导参数的权重,其取值范围为 $[0.333, 1]$, 且 $w_1 + w_2 = 1$.

由于权重的重要性,算法将权重作为待优化的参数,在协同进化过程中自适应调整.进化参数由进化算法实时给出,比例制导参数的计算如下:

追逃双方的相对位置为

$$x = x_w - x_m, \quad y = y_w - y_m \tag{10}$$

追逃双方的相对速度为

$$\begin{cases} V_x = V_w \cos \alpha_w - V_m \cos \alpha_m, \\ V_y = V_w \sin \alpha_w - V_m \sin \alpha_m. \end{cases} \tag{11}$$

追逃双方距离的平方为

$$r^2 = x^2 + y^2 \tag{12}$$

比例制导参数的估计值为

$$\tilde{\mu} = N_{PN} \dot{\alpha} \tag{13}$$

其中:制导常数 N_{PN} 的取值范围一般为 $[1.0, 6.0]$, $\dot{\alpha} = (\dot{x} v_y - \dot{y} v_x) / r^2$ 为视线角变化率.

考虑到追捕者运动受最小转弯半径的约束,启发式参数可修正为

$$\mu_{w_2} = \begin{cases} \tilde{\mu}, & |\tilde{\mu}| \leq 1; \\ \text{sgn}(\tilde{\mu}), & \text{其他} \end{cases} \tag{14}$$

4.2 适应度定义

在协同进化算法中,适应度通常定义为所有对抗结果的累加,这就是简单适应度. Rosin 和 Below^[10] 为竞争型协同进化提出了共享适应度,以保证协同进化“军备竞赛”的持续.在共享适应度中,每个个体都被当作一个独立的资源,由群体中可以击败它的所有对手共享.个体 i 的共享适应度为

$\frac{1}{\sum_{i \in X} N_i}$. 其中: X 为个体集合, N_i 为群体中所有击败个体 i 的对手的总和.

该方法的优点是:如能击败那些很难被同伴击败的对手,即使它击败的对手没有同伴多,该个体也能获得较高的奖赏.

4.3 协同进化算法流程

逃跑者采用混沌变异,追捕者在混沌、高斯、柯西和均匀变异中随机.选取编码采用浮点数编码,即每个基因用一个浮点数来表示选择.算子采用比例

选择算子,同时采取精英替代策略;交叉算子采用单点交叉算子.协同进化算法的过程描述如下:

步骤 1: 进化代数计数器初始化.

步骤 2: 随机产生 3 个初始子群体,即 1 个逃跑者子种群和 2 个追捕者子种群.

步骤 3: 逃跑者子种群的个体,分别在 2 个追捕者子种群中随机选取 5 个追捕者与之对抗,即逃跑者子种群的个体与追捕者对抗 500 次.每次对抗中,3 个个体按照各自的控制信号运动,若逃跑者进入两追捕者的捕获半径或对抗时间超过规定的时间,则对抗结束.然后追捕者子种群 1 和子种群 2 的各个体,分别在其他 2 个子种群中随机选取 5 个对手和合作者对抗 500 次,累计适应度.这样可保证每个个体至少参与 5 次对抗,根据其胜负情况和参与次数计算平均适应度和共享适应度.

步骤 4: 终止条件判断.若满足终止条件,则输出当前各种群的最优个体,算法结束;否则,继续以下进化过程:

- 1) 分别对子种群进行交叉操作;
- 2) 分别对子种群进行变异操作;
- 3) 分别对子种群进行选择 and 复制操作;
- 4) 根据步骤 3 描述的过程,分别评价 3 个子群体中各个体的平均适应度;
- 5) 根据评价适应度和对抗结果,计算各个体的共享适应度或混合适应度;
- 6) 根据共享适应度或混合适应度,保留各子种群的最优个体;
- 7) 累加进化代数,返回步骤 4.

5 仿真结果及分析

5.1 仿真结果

逃跑者的控制信号由进化过程给出,追捕者的控制信号按式(9)进化.逃跑者和追捕者首先根据与对手的多次对抗结果计算简单和共享适应度,并用进化算法调整自己的控制参数;然后随机选择不同的对手进行对抗,根据对抗结果调整控制参数.这样不断反复,以实现自身策略的进化.仿真参数如下:

$$a_w = 0.0275, \quad b_w = 0.1, \quad R_w = 2.0, \quad c_w = 0.1, \\ a_m = 0, \quad b_m = 0.1, \quad R_m = 1.0, \quad c_m = 0.25;$$

捕获半径为 1.0 m,逃跑者的初始位置为(30 m,0),2 个追捕者的初始位置为(0,0),追逃双方一开始以最大速度 10 m/s 和 5 m/s 运动,方向为 $[-90^\circ, 90^\circ]$ 内随机取值,制导系数为 4.0,对抗时间为 12 s.逃跑者进入 2 个追捕者的捕获半径或对抗时间超过 12 s,对抗结束.

各子种群大小均为 100,进化代数为 1000.逃跑



者进化出多种奇异的逃跑轨迹,如“几”、“8”和“反S”形等.在简单适应度下,第60,350,720和760代最优追逃个体之间的轨迹如图2~图5所示.其中:实线表示逃跑者的轨迹,虚线和点划线表示2个追捕者的轨迹,“o”和“*”分别表示起点和终点.

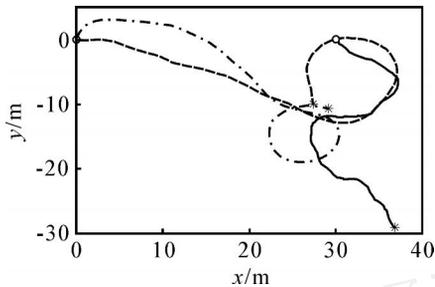


图2 第60代时最优追逃个体的轨迹

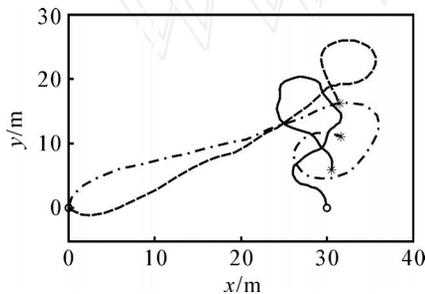


图3 第350代时最优追逃个体的轨迹

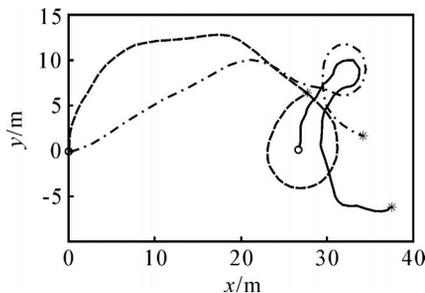


图4 第720代时最优追逃个体的轨迹

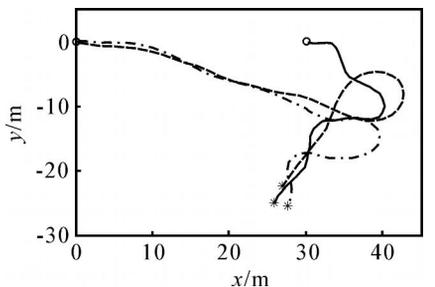


图5 第760代时最优追逃个体的轨迹

5.2 结果分析

5.2.1 与比例制导算法的比较

若式(9)中 $w_1 = 0$,则追捕者的算法退化为比例制导算法,逃跑者的策略针对追捕者的比例制导策略而进化.逃跑者子种群大小为20,逃跑者进化18代后,总能摆脱比例制导追捕者的追捕,即逃跑

者找到了规避比例制导追捕者的有效策略.在追逃对抗中,追捕者不能采取某种固定策略.本文提出的协同进化算法能提高追捕者的追捕能力.

5.2.2 误差分析

为追逃双方设置不同的初始运动方向,针对不同适应度进行20次仿真,每次仿真中进化1000代,统计各代子种群的最优个体之间的对抗结果.简单共享度下平均捕获200次,各次仿真的捕获次数分别为

251,197,161,107,167,180,138,266,94,119,
215,299,99,170,249,301,356,76,337,227.

共享适应度下平均捕获294次,各次捕获次数分别为

418,158,372,224,201,367,215,388,335,439,
299,306,196,332,265,382,293,226,228,238.

在追逃对抗中,追捕者需要不断测量逃跑者的速度,然后采取针对性的措施.

下面讨论速度测量误差对结果的影响.在仿真中,为速度的观测值加入白噪声 $N(0, \frac{2}{N})$,其中

$$N = s/10^{\text{SNR}/20}. \quad (15)$$

式中: $s = 0.206$ 为逃跑者速度的均方差,根据前5次进化过程中的速度计算得到;信噪比 SNR 取为10.

在有误差的情况下,简单适应度下平均捕获次数为284次,各次仿真的捕获次数分别为

267,345,291,301,463,306,281,167,300,283,
246,197,249,351,277,408,131,300,275,244.

在共享适应度下平均捕获277次,各次仿真捕获次数分别为

419,158,259,268,385,385,381,232,372,458,
275,150,242,300,221,247,181,267,269,274.

由于误差的扰动作用,简单适应度下捕获次数反而增加,竞争适应度下捕获次数下降不大,表明了该算法的稳健性.

5.2.3 时间复杂度分析

协同进化算法的核心是遗传算法,因此在该算法的基础上讨论协同进化算法的时间复杂度.根据4.3节的描述可知:协同进化算法需要对3个子种群进行遗传操作;在计算各个体的简单和共享适应度之前,需要对抗1500次;在每次对抗中,控制信号被离散为60步(相当于遗传算法的优化变量数).为了提高精度,算法在相邻2个控制信号之间进行4次线性插值,因此协同进化算法的时间复杂度是遗传算法(60个优化变量)的 $3 * 1500 * 4 = 18000$ 倍.

在 DotNet 2.0 框架下,用 C# 语言实现了该算法,运行环境为 Intel Dual 1.6 GHz CPU,1G内存,

Windows 2000 Server 操作系统. 简单适应度下程序运行 20 次的平均时间为 16 min 58 s, 共享适应度下为 17 min 6 s.

需要说明的是, 编程中未采取多线程技术, 因此在运算中 CPU 只有一个核工作, 但在命令行编译时开启了优化选项“/o”.

6 结 论

本文阐述了协同进化追逃的理论基础, 对二维平面追逃问题进行描述, 建立了动力学约束下的追逃运动数学模型. 该模型考虑了最小转弯半径, 在小角度机动时逃跑者能进行加速, 因此更符合实际情况. 协同进化仿真结果表明, 进化的逃跑策略能有效规避比例制导的追捕者, 逃跑者涌现出复杂多变的策略. 下一步将重点研究决策与行动之间存在时延情况下的追逃对抗, 以及分析具体追逃策略的对抗能力.

参考文献(References)

[1] Isaacs R. Differential games [M]. New York: John Wiley and Sons, 1965.

[2] 王泉德. 机器学习及其在多 Agent 对策学习中的应用研究[D]. 武汉: 武汉大学, 2005.
(Wang Q D. Machine learning and its application in multi-agent game learning [D]. Wuhan: Wuhan University, 2005.)

[3] Hillis W D. Coevolving parasites improve simulated evolution as an optimization procedure [C]. Artificial Life . Redwood City: Addison-Wesley, 1992: 313-324.

[4] Miller G F, Cliff D. Co-evolution of pursuit and evasion : Biological and game-theoretic foundations [R]. Brighton: University of Sussex, 1994.

[5] Reynolds C W. Competition, coevolution and the game of tag [C]. Proc of the 4th Int Workshop on the Synthesis and Simulation of Living Systems. Cambridge: MIT Press, 1994, 4: 59-69.

[6] Tahk M J, Sun B C. Co-evolutionary augmented lagrangian methods for constrained optimization [J]. IEEE Trans on Evolutionary Computation, 2000, 4 (2): 114-124.

[7] Choi H L, Bang H C, Tahk M J. Co-evolutionary optimization of three-dimensional target evasive maneuver against a proportionally guided missile [C]. Proc of the 2001 Congress on Evolutionary Computation. New York: IEEE, 2001, 2: 1406-1413.

[8] 关志华, 寇纪淞, 李敏强. 基于 λ -约束方法的增广 Lagrangian 多目标协同进化算法 [J]. 系统工程与电子技术, 2002, 24(9): 33-37.
(Guan Z H, Kou J S, Li M Q. A coevolutionary algorithm based on λ -constraint and augmented Lagrangian methods[J]. Systems Engineering, 2002, 24 (9): 33-37.)

[9] Kim J G, Tahk M J. Co-evolutionary computation for constrained min-max problems and its applications for pursuit-evasion games [C]. Proc of Congress on Evolutionary Computation. Seoul, 2001, 2: 1205-1212.

[10] Rosin C D, Belew R K. New methods for competitive coevolution [J]. Evolutionary Computation, 1997, 5 (1): 1-29.

下 期 要 目

动态零等待流水线调度问题的滚动策略及优化算法	钱 斌, 等
针对时间距离约束的网络化控制系统带宽调度策略	康 军, 等
一种基于关联规则的多类标分类算法	李 宏, 等
基于局部均值与类均值的近邻分类	曾 勇, 等
基于 k -最近邻的支持向量预选取方法	韩德强, 等
进化强化学习及其在机器人路径跟踪中的应用	段 勇, 等
基于运动预测的路径跟踪最优控制	武 星, 楼佩煌
一类离散不确定时滞系统的输出变结构控制	颜闽秀, 等
梯度粒子群优化算法及其收敛性分析	肖健梅, 等
基于改进在线支持向量回归的离子浓度预测模型	王凌云, 等