Journal of System Simulation

Volume 32 | Issue 5 Article 3

5-15-2020

Optimization Algorithm Based on Human Infection with Avian Influenza

Guangqiu Huang
School of Management, Xi'an University of Architecture and Technology, Xi'an 710055, China;

Qiuqin Lu
School of Management, Xi'an University of Architecture and Technology, Xi'an 710055, China;

Optimization Algorithm Based on Human Infection with Avian Influenza

Abstract

Abstract: To get the global optimal solution of some complex nonlinear optimization problems, *an optimization algorithm based on the human avian influenza infectious diseases is proposed by using its dynamic model of cross species transmission. Applies the H7N9 infectious disease model to create the operators S^u-S^u, I^u-I^u, S^u-I^u, I^u-S^u, S^u-D^u, I^u-D^u, and to enable the individuals to exchange information among the same species and cross-species. The S^u-S^u and I^u-I^u operators can improve the characteristics of the weak individuals by that of the strong individuals, thus the exploitation ability of the algorithm can be improved. The S^u-I^u and I^u-S^u operators can improve the fitness distribution characteristics of the individuals, therefore, the exploration ability of the algorithm can be improved. The S^u-D^u and I^u-D^u operators can effectively remove the extremely weak individual, thereby the probability that the algorithm falls into a local trap can be reduced*. The test cases show that the algorithm can quickly solve some complex nonlinear optimization problems with high dimensions.

Keywords

swarm intelligent optimization algorithm, bin model, epidemic dynamics

Recommended Citation

Huang Guangqiu, Lu Qiuqin. Optimization Algorithm Based on Human Infection with Avian Influenza[J]. Journal of System Simulation, 2020, 32(5): 767-781.

第 32 卷第 5 期 系统仿真学报© Vol. 32 No. 5
2020 年 5 月 Journal of System Simulation May, 2020

人感禽流感优化算法

黄光球，陆秋琴
（西安建筑科技大学管理学院，陕西 西安 710055）

摘要：为求解一类复杂非线性优化问题的全局最优解，采用跨物种传播的人感禽流感传染病动力学模型提出了人感禽流感传染病优化算法。利用 H7N9 传染病模型构造出的 S^u-S^u，I^u-I^u，S^u-I^u，I^u-S^u，S^u-D^u，I^u-D^u 等算子能使个体能在同物种和跨物种个体之间充分交换信息，其中 S^u-S^u、I^u-I^u 算子可利用强壮个体的特征来改善虚弱个体的特征，从而提升算法的求精能力；S^u-I^u、I^u-S^u 算子可改良个体的适应度分布特征，从而提升算法的探索能力；S^u-D^u，I^u-D^u 算子可使极虚弱个体得到有效清除，从而降低算法陷入局部陷阱的概率。测试案例表明：本算法可快速求解一类维数较高的复杂非线性优化问题。

关键词：群智能优化算法；仓室模型；传染病动力学
中图分类号：TP391.9 文献标识码：A 文章编号：1004-731X (2020) 05-0767-15
DOI: 10.16182/j.issn1004731x.joss.19-0077

Optimization Algorithm Based on Human Infection with Avian Influenza

Huang Guangqiu, Lu Qiuqin
(School of Management, Xi'an University of Architecture and Technology, Xi'an 710055, China)

Abstract: To get the global optimal solution of some complex nonlinear optimization problems, *an optimization algorithm based on the human avian influenza infectious diseases is proposed by using its dynamic model of cross species transmission. Applies the H7N9 infectious disease model to create the operators S^u-S^u, I^u-I^u, S^u-I^u, I^u-S^u, S^u-D^u, I^u-D^u, and to enable the individuals to exchange information among the same species and cross-species. The S^u-S^u and I^u-I^u operators can improve the characteristics of the weak individuals by that of the strong individuals, thus the exploitation ability of the algorithm can be improved. The S^u-I^u and I^u-S^u operators can improve the fitness distribution characteristics of the individuals, therefore, the exploration ability of the algorithm can be improved. The S^u-D^u and I^u-D^u operators can effectively remove the extremely weak individual, thereby the probability that the algorithm falls into a local trap can be reduced*. The test cases show that the algorithm can quickly solve some complex nonlinear optimization problems with high dimensions.

Keywords: swarm intelligent optimization algorithm; bin model; epidemic dynamics

引言

工程中存在大量高度非线性优化问题，此类问题具有大量局部最优解，传统的基于梯度的优化算法无法求解该类问题[1]。目前，此类问题的求解方法是群智能算法[2]。常见的群智能优化算法有 GA[3]，PSO[4]，BBO[5]，DE[6]，AIA[7]，AFSA[8]等。这类算法的共同特点是算法所依赖的生物场景很简单，可开发出来的算子较少。

迄今为止，绝大多数群智能优化算法均源于大

收稿日期：2019-02-28 修回日期：2020-03-25；
基金项目：国家自然科学基金(71874134)，陕西省自然科学基础研究计划-重点项目(2019JZ-30)，陕西省社会科学基金(2018S49，2017S035)；
作者简介：黄光球(1964-)，男，湖南桃源，博士，教授，研究方向为计算机仿真、计算智能。

自然中的一些特殊生物进化场景带给我们的启示。优化问题的每个试探解比喻成大自然中具有生物特征的个体，生物个体之间的相互作用关系用于开发群智能算法的算子与逻辑结构。如果一个生物进化场景具有特殊性质，那么该场景会给群智能优化算法的设计和分析带来很大方便。能够跨物种传播的禽流感病毒攻击禽类和人类的过程，就是这样一个场景。

人感禽流感是由禽流感病毒引起的人类疾病[9]。由于禽流感病毒的血凝素结构等特点，一般感染禽类，当病毒在复制过程中发生基因重配，致使结构发生改变，获得感染人的能力，就可能造成人感禽流感疾病的发生[10]。至今发现能直接感染人的禽流感病毒亚型有：H5N1，H7N1，H7N2，H7N3，H7N7，H9N2 和 H7N9 亚型。其中，高致病性 H5N1 亚型和 H7N9 亚型尤为引人关注，不仅造成了人类的伤亡，同时可重创家禽养殖业[11]。

依据 Kermack-Mckendrick 仓室建模方法[12-13]而构建的人感禽流感传染病动力学模型是描述人类在禽流感传染病毒作用下其动态行为在易感、感染、死亡等状态之间进行随机转换的一种非线性数学模型，该模型不是从病理知识的角度考虑传染病，而是按照一般传染病传播机理通过数量关系描述传染病的传播过程、分析感染个体数的变化规律、揭示传染病的发展性态。

禽流感传染病毒在不同物种及其个体之间的传播，使得不同物种及其个体之间的相互作用关系体现得很充分，其生物学含义明确；禽流感传染病毒攻击的是禽类和人类个体的少部分器官，将该现象映射到对优化问题的求解，就是每次处理的变量个数只是全部变量的极少部分。因此，算法中变量处理策略的生物学含义相当明确。

本文以禽流感传染病毒攻击禽类和人类这一场景为依托，提出了一种新的群智能优化算法，称其为 H7N9 算法，该算法既能利用禽流感病毒传播来体现个体之间的明确的相互作用关系，又能确保每次只处理优化问题的极少部分变量，从而大幅提升算法的计算速度，且整个算法又具有良好的数学模型基础，从而为算法的性能分析提供方便。本文着重解决了如下 5 个问题：

(1) 如何将 H7N9 传染病动力学模型转化为能求解复杂优化问题的群智能优化算法；

(2) 如何使得 H7N9 算法中的算子能充分反映不同物种个体之间的相互作用关系，以便体现 H7N9 传染病动力学模型的思想；

(3) 如何证明 H7N9 算法的全局收敛性；

(4) 如何确定 H7N9 算法的最佳参数设置；

(5) 如何进行 H7N9 算法的动态行为分析。

1 H7N9 优化算法设计原理

设要求解的优化问题为

$$\begin{cases}\min F(\boldsymbol{X}) \\ \text{s.t.} \quad \boldsymbol{X}\in H\subset R^n\end{cases} \tag{1}$$

式中：R^n 为 n 维欧氏空间；$\boldsymbol{X}=(x_1, x_2, \ldots, x_n)$为一个 n 维决策向量；H 为搜索空间；$F(\boldsymbol{X})$为目标函数。

1.1 算法场景设计

在一个小镇有一个小型鸡禽养殖场和一个电器制造厂。养殖场养殖了 N_1 只鸡，其编号是 1，2，…，N_1。这些鸡专门供电器制造厂的工人食用，工人的编号是 1，2，…，N_2。每只鸡或每个工人，均由 n 个特征(器官)来表征。令 u 表示个体类型，u=1 表示鸡类，u=2 表示工人类，即对类型为 u 的个体 i 来说，其表征特征为 $(x_{i,1}^u, x_{i,2}^u, \cdots, x_{i,N_u}^u)$。

该小镇有一种称为 H7N9 的传染病在鸡群之中流行。H7N9 具有很强的传染能力，既能够在同物种类内传播，又能够跨物种传播。H7N9 首先在鸡群中传播，鸡染上 H7N9 后，一部分能够治愈并复原，另一部分则死亡。当工人食用已染病的鸡后，就会传染上 H7N9。当 H7N9 由鸡传染给工人之后，还能够在工人之中继续传播；工人染上 H7N9 后，一部分能够治愈并复原，另一部分则死亡。H7N9 攻击的是鸡和工人的部分特征(器官)。

将上述场景映射到对优化问题式(1)全局最优

解的搜索过程中，含义如下所述。

优化问题式(1)的搜索空间与小镇相对应，该小镇中一只鸡和一名工人分别对应于优化问题式(1)的一个试探解，即 N_1 只鸡所对应的试探解集就是 $\{\boldsymbol{X}_1^1,\boldsymbol{X}_2^1,\cdots,\boldsymbol{X}_{N_1}^1\}$，$N_2$ 名工人所对应的试探解集就是 $\{\boldsymbol{X}_1^2,\boldsymbol{X}_2^2,\cdots,\boldsymbol{X}_{N_2}^2\}$，且类型为 u 的个体 i 的特征 j 与试探解 $\boldsymbol{X}_{N_u}^u$ 的变量 $x_{i,j}^u$ 相对应。对于优化问题式(1)，类型为 u 的个体 i 的适应度 Fit 按式(2)计算：

$$Fit(\boldsymbol{X}_i^u)=\begin{cases}\dfrac{1}{1+F(\boldsymbol{X}_i^u)}, & \text{若 } F(\boldsymbol{X}_i^u)\geqslant 0\\ 1+\left|F(\boldsymbol{X}_i^u)\right|, & \text{若 } F(\boldsymbol{X}_i^u)<0\end{cases} \tag{2}$$

1.2 可跨物种传播的传染病动力学模型

该小镇中共有六类群体：易感(未染病)的鸡，其在时期 t 的占比为 $S^1(t)$，此类种群用 S^1 表示；已染病的鸡，其在时期 t 的占比为 $I^1(t)$，此类种群用 I^1 表示；已死亡的鸡，其在时期 t 的占比为 $D^1(t)$，此类种群用 D^1 表示；易感(未染病)的工人，其在时期 t 的占比为 $S^2(t)$，此类种群用 S^2 表示；已染病的工人，其在时期 t 的占比为 $I^2(t)$，此类种群用 I^2 表示；已死亡的工人，其在时期 t 的占比为 $D^2(t)$，此类种群用 D^2 表示。上述场景可采用传染病传播仓室模型[14]来描述，如图 1 所示。

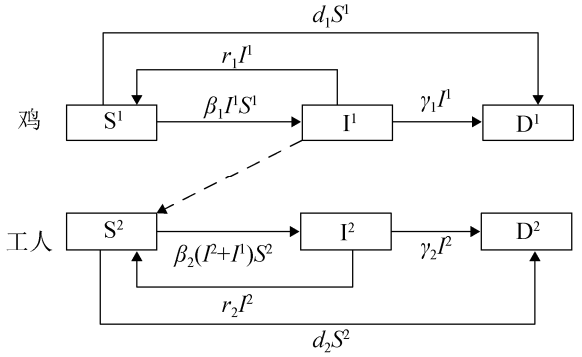


图 1 鸡与工人相互作用的传染病传播流程图

Fig. 1 Infectious disease transmission flow chart of chicken worker interaction

根据图 1，我们可以写出其相应的动力学方程组，如式(3)所示：

$$\begin{cases}\dfrac{\mathrm{d}S^1}{\mathrm{d}t}=-d_1S^1-\beta_1I^1S^1+D^1+r_1I^1\\ \dfrac{\mathrm{d}I^1}{\mathrm{d}t}=\beta_1I^1S^1-\gamma_1I^1-r_1I^1\\ \dfrac{\mathrm{d}D^1}{\mathrm{d}t}=d_1S^1+\gamma_1I^1-D^1\\ \dfrac{\mathrm{d}S^2}{\mathrm{d}t}=-d_2S^2-\beta_2(I^2+I^1)S^2+D^2+r_2I^2\\ \dfrac{\mathrm{d}I^2}{\mathrm{d}t}=\beta_2(I^2+I^1)S^2-\gamma_2I^2-r_2I^2\\ \dfrac{\mathrm{d}D^2}{\mathrm{d}t}=d_2S^2+\gamma_2I^2-D^2\end{cases} \tag{3}$$

式中：t 为时期；β_1 为 H7N9 在鸡群中的传染率，$0<\beta_1<1$；d_1 为鸡的自然死亡率，$0<d_1<1$；r_1 为鸡染病后的治愈率，$0<r_1<1$；γ_1 为鸡因染上 H7N9 而死亡的死亡率，$0<\gamma_1<1$；β_2 为 H7N9 由鸡向工人传播的传播率，$0<\beta_2<1$；d_2 为工人的自然死亡率，$0<d_2<1$；r_2 为工人染病后的治愈率，$0<r_2<1$；γ_2 为因染上 H7N9 而死亡的死亡率，$0<\gamma_2<1$。

为简单起见，对上述场景进行一些简化：当一只鸡自然死亡或因染上 H7N9 而死亡后，立即就有一只新鸡被添加到养殖场，当一名工人自然死亡或因染上 H7N9 而死亡后，立即就有一名新工人被招聘到电器制造厂，从而确保鸡的总数 N_1 和工人总数 N_2 为常数。

$S^1(t)$，$I^1(t)$，$D^1(t)$，$S^2(t)$，$I^2(t)$，$D^2(t)$重要的含义是：对于任意一只鸡来说，$S^1(t)$，$I^1(t)$，$D^1(t)$分别表示该鸡属于 S^1 类、I^1 类、D^1 类的概率，或者说该鸡分别处于 S^1 状态、I^1 状态、D^1 状态的概率；类似地，对于任意一名工人来说，$S^2(t)$，$I^2(t)$，$D^2(t)$分别表示该工人属于 S^2 类、I^2 类、D^2 类的概率，或者说该工人分别处于 S^2 状态、I^2 状态、D^2 状态的概率。

必须指出，在同一时间点，一只鸡只能处在 S^1 状态、I^1 状态、D^1 状态中的某一个；同理，一名工人也只能处在 S^2 状态、I^2 状态、D^2 状态中的某一个。

通常情况下模型式(3)中的参数 d_1，β_1，γ_1，r_1，d_2，β_2，γ_2，r_2 的取值是随时间变化的，将式(3)应用到任一只鸡 i 和任一名工人 i 上，并将式(3)改写

成离散递推形式，则有：

$$\begin{cases} S_i^1(t+1)=S_i^1(t)-d_1S_i^1(t)-\beta_1I_i^1(t)S_i^1(t)+ \\ \qquad D_i^1(t)+r_1I_i^1(t) \\ I_i^1(t+1)=I_i^1(t)+\beta_1I_i^1(t)S_i^1(t)-\gamma_1I_i^1(t)-r_1I_i^1(t) \\ D_i^1(t+1)=1-S_i^1(t+1)-I_i^1(t+1) \\ S_i^2(t+1)=S_i^2(t)-d_2S_i^2(t)-\beta_2(I_i^2(t)+ \\ \qquad I_i^1(t))S_i^2(t)+D_i^2(t)+r_2I_i^2(t) \\ I_i^2(t+1)=I_i^2(t)+\beta_2(I_i^2(t)+I_i^1(t))S_i^2(t)- \\ \qquad \gamma_2I_i^2(t)-r_2I_i^2(t) \\ D_i^2(t+1)=1-S_i^2(t+1)-I_i^2(t+1) \end{cases} \tag{4}$$

1.3 个体演化状态识别

在时期 t，采用模型式(4)计算鸡 i 的易感概率 $S_i^1(t)$、染病概率 $I_i^1(t)$ 和死亡概率 $D_i^1(t)$，以及工人 i 的易感概率 $S_i^2(t)$、染病概率 $I_i^2(t)$ 和死亡概率 $D_i^2(t)$。鸡 i 在时期 t 处于 S^1 状态、I^1 状态和 D^1 状态中的哪个状态，由 $S_i^1(t)$，$I_i^1(t)$，$D_i^1(t)$ 中的最大者确定。同理，工人 i 在时期 t 处于 S^2 状态、I^2 状态和 D^2 状态中的哪个状态，由 $S_i^2(t)$，$I_i^2(t)$，$D_i^2(t)$ 中的最大者确定。依据图 1，可以识别出所有合法的状态转移类型，如图 2 所示，图 2 中所描述的状态转移类型的含义及其所对应的算子如表 1 所示。

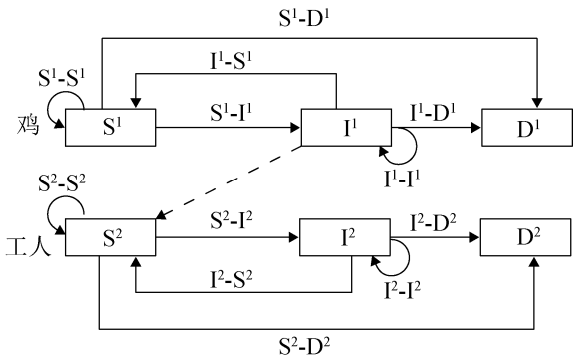


图 2 合法的状态转移类型
Fig. 2 Legal state transition types

特别注意，表 1 中的状态转移 $S^1\rightarrow D^1$ 和 $I^1\rightarrow D^1$ 以及 $S^2\rightarrow D^2$ 和 $I^2\rightarrow D^2$ 分别表示鸡和工人自然死亡和因染上 H7N9 而死亡。

表 1 合法状态转换
Tab. 1 Legal state transitions

个体类型	在时期 t 的状态	在时期 t+1 的状态	状态转换	算子
鸡	S^1	S^1	$S^1\rightarrow S^1$	S^1-S^1
	S^1	I^1	$S^1\rightarrow I^1$	S^1-I^1
	S^1	D^1	$S^1\rightarrow D^1$	S^1-D^1
	I^1	I^1	$I^1\rightarrow I^1$	I^1-I^1
	I^1	S^1	$I^1\rightarrow S^1$	I^1-S^1
	I^1	D^1	$I^1\rightarrow D^1$	I^1-D^1
工人	S^2	S^2	$S^2\rightarrow S^2$	S^2-S^2
	S^2	I^2	$S^2\rightarrow I^2$	S^2-I^2
	S^2	D^2	$S^2\rightarrow D^2$	S^2-D^2
	I^2	I^2	$I^2\rightarrow I^2$	I^2-I^2
	I^2	S^2	$I^2\rightarrow S^2$	I^2-S^2
	I^2	D^2	$I^2\rightarrow D^2$	I^2-D^2

从表 1 可知，H7N9 算法共有 12 个算子，较多的算子对本算法的性能提升是否有益？下面给出定理 1 来回答此问题。

定理 1 若一个群智能优化算法所拥有的算子越多，且各算子被随机独立调度执行，则该算法的性能越优良。

证：假设一个群智能优化算法有 n 个算子，当该算法求解优化问题 X 时，这 n 个算子求解优化问题 X 成功的概率分别为 p_1，p_2，…，p_n。因每个算子在求解优化问题 X 时均是被随机独立调度执行的，故该算法在其 n 个算子的联合作用下求解优化问题 X 成功的概率 q 为

$$q=1-\prod_{i=1}^{n}(1-p_i)$$

因 $0<p_i<1$，故 $0<1-p_i<1$，$i=1\sim n$；当 n 越大时，$\prod_{i=1}^{n}(1-p_i)$ 越小，而 q 则越大。此结论表明，若一个群智能优化算法的算子越多，则该算法求解一个优化问题时成功的概率越大。因此，算子越多，算法性能越优良。

H7N9 算法拥有 12 个算子，只要确保每个算子均是被随机调度执行的，就能确保 H7N9 算法满足定理 1 的条件，从而使其具有优良的性能。此结论为 H7N9 算法的架构设计指明了方向。

DOI: 10.16182/j.issn1004731x.joss.19-0077

1.4 演化算子设计

1.4.1 特征集合生成方法

设当前个体类型 $u\in\{1，2\}$，个体编号为 i，个体状态 $s\in\{S^u，I^u\}$，则

(1) 强壮个体集合 PS_s^u 的产生方法：从类型为 u 且处于状态 s 的个体中随机挑出 L 个个体，这些个体的适应度是类型为 u 的个体中最高的，形成强壮个体集合 PS_s^u；L 称为施加影响的个体数。

(2) 普通个体集合 CS_s^u 的产生方法：从类型为 u 且处于状态 s 的个体中随机挑出 L 个个体，形成普通个体集合 CS_s^u。

(3) 虚弱个体集合 WS_s^u 的产生方法：从类型为 u 且处于状态 s 的个体中随机挑出 L 个个体，这些个体的适应度是类型为 u 的个体中最小的，形成虚弱个体集合 WS_s^u。

1.4.2 状态转移算子设计方法

设当前个体类型 $u\in\{1，2\}$，个体编号为 i，个体状态 $s\in\{S^u，I^u，D^u\}$，则

(1) S^u-S^u 算子。该算子描述的是在时期 t 类型为 u 且处于状态 S^u 的易感个体，在时期 t+1 仍处于 S^u 状态(易感状态)的情形。依据达尔文进化论，生物个体在生存竞争过程中总是尽量使自身强壮，以便更好生存发展。为达到此目的，可将强壮个体集合 $PS_{S_u}^u$ 中的 L 个易感个体的特征 j 及其状态值经加工处理后传给易感个体 i 的对应特征 j，使其具有与 $PS_{S_u}^u$ 中强壮个体的特征。即在时期 t+1，对处于 S^u 状态的易感个体 i，有

$$v_{i,j}^u(t+1)=\sum_{k\in PS_{S_u}^u}[\alpha_k x_{k,j}^u(t)+\beta_k x_{k,j}^u(t)] \tag{5}$$

式中：α_k 和 β_k 为常数，依据文献[6]，可取 α_k=*Rand*(0.7，0.9)，β_k=*Rand*(-0.5，0.5)；*Rand*(a，b)为在区间[a，b]内产生均匀分布随机数。

(2) I^u-I^u 算子。该算子的特征与 S^u-S^u 算子类似，其差别在于个体所处的状态均是 I^u，即

$$v_{i,j}^u(t+1)=\sum_{k\in PS_{I_u}^u}[\alpha_k x_{k,j}^u(t)+\beta_k x_{k,j}^u(t)] \tag{6}$$

(3) S^1-I^1 算子。该算子描述的是在时期 t 处于 S^1 状态的易感鸡，通过与已染病的其他 I^1 类鸡接触后染上 H7N9 的情形。因 H7N9 可以在鸡群之中传播，故将 $CS_{I_1}^1$ 中 L 只已染病鸡的特征 j 及其状态值经加工处理后传给易感鸡 i 的对应特征 j，使其染上 H7N9。即在时期 t+1，对处于 S^1 状态的易感鸡 i，有

$$v_{i,j}^1(t+1)=\begin{cases}x_{i_1,j}^1(t)+x_{i_2,j}^1(t)-x_{i_3,j}^1(t) & 若L\geqslant 3\\ 0.5(x_{i_1,j}^1(t)+x_{i_2,j}^1(t)) & 若L=2\\ x_{i_1,j}^1(t) & 若L=1\\ x_{i,j}^1(t) & 若L=0\end{cases} \tag{7}$$

式中：i_1，i_2 和 i_3 分别从集合 $CS_{I_1}^1$ 中随机选择，且 $i_1\neq i_2\neq i_3$。

(4) S^2-I^2 算子。该算子描述的是在时期 t 处于 S^2 状态的易感工人，通过食用已染病的 I^1 类鸡或接触已染病的 I^2 类工人而染上 H7N9 的情形。将 $CS_{I_1}^1$ 中 L 个已染病鸡和 $CS_{I_2}^2$ 中 L 个已染病的工人、的特征 j 及其状态值经加工处理后传给易感工人 i 的对应特征 j，使其染上 H7N9。即在时期 t+1，对处于 S^2 状态的易感工人 i，有

$$v_{i,j}^2(t+1)=\sum_{k\in CS_{I_1}^1}\alpha_k x_{k,j}^1(t)+\sum_{k\in CS_{I_2}^2}\beta_k x_{k,j}^2(t) \tag{8}$$

(5) I^u-S^u 算子。该算子描述的是在时期 t 类型为 u 且处于 I^u 状态的已染病个体，通过治疗而治愈后重新转变为易感的 S^u 状态的情形。将 $CS_{S_u}^u$ 中 L 个易感个体的特征 j 及其状态值经加工处理后传给已染病的个体 i 的对应特征 j，使其具有易感特征的个体。即在时期 t+1，对处于 I^u 状态的已染病个体 i，有

$$v_{i,j}^u(t+1)=\begin{cases}x_{i_1,j}^u(t)+0.5(x_{i_2,j}^u(t)-x_{i_3,j}^u(t)) \\ \quad Rand(0,1)<0.9或Rand(1,n)=j \\ x_{i_1,j}^u(t) \quad 其他\end{cases} \tag{9}$$

式中：i_1，i_2 和 i_3 分别从集合 $CS_{S_u}^u$ 中随机选择，且 $i_1\neq i_2\neq i_3$。

(6) S^u-D^u 算子。该算子描述的是在时期 t 类型为 u 且处于 S^u 状态的易感个体，因自然死亡后而重新产生一个新个体这一情形。在 $WS_{S_u}^u$ 中任选一

个体 z，令该个体死亡，也即将该个体的信息删除掉，然后将该个体 z 替换成一个从易感的强壮个体集合 $PS_{S_u}^u$ 中任意挑选出来的易感个体 w。即在时期 t+1，对处于 S^u 状态的易感个体 z，有

$$\boldsymbol{X}_z^u(t+1)=\boldsymbol{X}_w^u(t) \qquad z\in WS_{S_u}^u, w\in PS_{S_u}^u \tag{10}$$

(7) I^u-D^u 算子。该算子描述的是在时期 t 类型为 u 且处于 I^u 状态的已染病个体，因染病死亡后而重新产生一个新个体的情形。在 $WS_{I_u}^u$ 中任选一个体 z，令该个体死亡，也即将该个体的信息删除掉，然后将该个体 z 替换成一个从已染病的强壮个体集合 $PS_{I_u}^u$ 中任意挑选出来的染病个体 w。即在时期 t+1，对处于 I^u 状态的已染病个体 z，有

$$\boldsymbol{X}_z^u(t+1)=\boldsymbol{X}_w^u(t) \qquad z\in WS_{I_u}^u, w\in PS_{I_u}^u \tag{11}$$

(8) 生长算子。对于优化问题式(1)，其生长算子可以描述为：

$$\boldsymbol{X}_i^u(t+1)=\begin{cases}\boldsymbol{V}_i^u(t+1) & \text{若} Fit(\boldsymbol{V}_i^u(t+1))>Fit(\boldsymbol{X}_i^u(t))\\ \boldsymbol{X}_i^u(t) & \text{其他}\end{cases},\ i=1\sim N \tag{12}$$

式中：$\boldsymbol{X}_i^u(t+1)=(x_{i,1}^u(t+1),x_{i,2}^u(t+1),\cdots,x_{i,N_u}^u(t+1))$，$\boldsymbol{V}_i^u(t+1)=(v_{i,1}^u(t+1),v_{i,2}^u(t+1),\cdots,v_{i,N_u}^u(t+1))$；函数 $Fit(\boldsymbol{V}_i^u(t+1))$ 和 $Fit(\boldsymbol{X}_i^u(t))$ 按式(2)计算。

1.5 H7N9 算法构造方法

(1) 初始化：a) 令时期 t=0，按第 3 节介绍的方法设置本算法中的参数：演化时期数 G，全局最优解误差 ε，个体特征被 H7N9 攻击的最大概率 E_0，L，N_1、N_2；b) 分别初始化 N_1 只鸡和 N_2 名工人：$\{\boldsymbol{X}_1^u(0),\boldsymbol{X}_2^u(0),\cdots,\boldsymbol{X}_{N_u}^u(0)\}$ b，$u\in\{1，2\}$。

(2) 产生 6 个随机数：$a_j^i=Rand(0，1)$，$b_j^i=Rand(0，1)$，j=1~3；计算 $A=\sum_{j=1}^{3}a_j^i$，$B=\sum_{j=1}^{3}b_j^i$，$S_i^1(0)=a_1^i/A$，$S_i^2(0)=b_1^i/B$，$I_i^1(0)=a_2^i/A$，$I_i^2(0)=b_2^i/B$，$D_i^u(0)=1-S_i^u(0)-I_i^u(0)$，$i$=1~$N$，$u\in\{1，2\}$。

(3) 计算类型为 u 的个体 i 的 S^u，I^u，D^u 状态，$SID_i^u(0)=GetSID\{S_i^u(0),I_i^u(0),D_i^u(0)\}$，$i$=1~$N$；// 函数 $GetSID$()用于确定类型为 u 的个体 i 将处于 S^u 状态、I^u 和 D^u 状态中的哪一个状态。

(4) 执行下列操作：

FOR t=1 **TO** G

按第 1.2 节介绍的方法确定 d_1，d_2，β_1，β_2，γ_1，γ_2，r_1，r_2；

FOR u=1 **TO** 2

FOR i=1 **TO** N_u

利用式(4)计算 $S_i^u(t)$、$I_i^u(t)$、$D_i^u(t)$；

计算 $SID_i^u(t)=GetSID(S_i^u(t),I_i^u(t),D_i^u(t))$；

FOR j=1 **TO** n

p=$Rand$(0，1)；

IF $p\le E_0$ **THEN**

IF $SID_i^u(t-1)=\mathrm{S}^u$ **THEN**

IF $SID_i^u(t)=\mathrm{S}^u$ **THEN**

利用 S^u-S^u 算子式(5)计算 $v_{i,j}^u(t)$；

$SID_i^u(t)=\mathrm{S}^u$；

ELSE IF $SID_i^u(t)=\mathrm{I}^u$ **THEN**

利用 S^u-I^u 算子式(7)或式(8)计算 $v_{i,j}^u(t)$；

$SID_i^u(t)=\mathrm{I}^u$；

ELSE IF $SID_i^u(t)=\mathrm{D}^u$ **THEN**

利用 S^u-D^u 算子式(10)产生新个体 z；

$SID_z^u(t)=\mathrm{S}^u$；

ELSE

$v_{i,j}^u(t)=x_{i,j}^u(t-1)$，

$SID_i^u(t)=SID_i^u(t-1)$；

END IF

ELSE IF $SID_i^u(t-1)=\mathrm{I}^u$ **THEN**

IF $SID_i^u(t)=\mathrm{I}^u$ **THEN**

利用 I^u-I^u 算子式(6)计算 $v_{i,j}^u(t)$；

$SID_i^u(t)=\mathrm{I}^u$；

ELSE IF $SID_i^u(t)=\mathrm{S}^u$ **THEN**

利用 I^u-S^u 算子式(9)计算 $v_{i,j}^u(t)$；

$SID_i^u(t)=\mathrm{S}^u$；

ELSE IF $SID_i^u(t)=\mathrm{D}^u$ **THEN**

利用 I^u-D^u 算子式(11)产生新个体 z；

$SID_z^u(t)=\mathrm{I}^u$；

DOI: 10.16182/j.issn1004731x.joss.19-0077

ELSE

$v_{i,j}^{u}(t)=x_{i,j}^{u}(t-1)$;

$SID_i^u(t)=SID_i^u(t-1)$;

END IF

END IF

ELSE

$v_{i,j}^{u}(t)=x_{i,j}^{u}(t-1)$;

$SID_i^u(t)=SID_i^u(t-1)$;

END IF

END FOR

按式(12)计算生长算子;

END FOR

END FOR

$\boldsymbol{X}(t)$=min($\boldsymbol{X}^1(t)$，$\boldsymbol{X}^2(t)$)；//$\boldsymbol{X}^1(t)$，$\boldsymbol{X}^2(t)$分别为鸡和工人个体的当前全局最优解

IF $|\boldsymbol{X}(t)-\boldsymbol{X}^*(t)|\le\varepsilon$ **THEN**//$\boldsymbol{X}^*(t)$为优化问题的当前全局最优解

转步骤(5);

END IF

$\boldsymbol{X}^*(t)=\boldsymbol{X}(t)$;

END FOR

(5) 结束。

函数 *GetSID*(p_S，p_I，p_D)的定义如下：

FUNCTION *GetSID*(p_S，p_I，p_D)//p_S，p_I，p_D 分别表示状态易感状态 S、已染病状态 I、死亡状态 D 的概率

p=*Rand*(0，1);

IF $p\le p_S$ **THEN**

RETURN S；//返回易感状态 S;

ELSE IF $p_S<p\le p_S+p_I$ **THEN**

RETURN I；//返回已染病状态 I;

ELSE

RETURN D；//返回死亡状态 D;

END IF

END FUNCTION

1.6 H7N9 算法的特性

(1) 优良性能。从 1.5 节描述算法结构知，H7N9 算法的每个算子均是被随机独立调度执行的，满足定理 1 的条件，因而 H7N9 算法具有优良的性能。

(2) Markov 特性。从 S^u-S^u，I^u-I^u，S^u-I^u，I^u-S^u，S^u-D^u，I^u-D^u 的定义知，新一代试探解的产生只与该试探解的当前状态有关，而与该试探解以前是如何演变到当前状态的历程无关，因而 H7N9 算法演化过程具有 Markov 特性。

(3)“步步不差”特性。从生长算子的定义可知，新一代试探解的适应度不会劣于其老一代试探解的适应度。

(4) 全局收敛性。H7N9 算法的全局收敛性证明可参见文献[8]。

1.7 时间复杂度

令 $N=N_1+N_2$，H7N9 算法的时间复杂度计算过程如表 2 所示。

表 2 H7N9 算法的时间复杂度计算表

Tab. 2 Time complexity calculation of H7N9 algorithm

操作	时间复杂度	最多循环次数
初始化	$O(3n+7(n+1)N+n^2N)$	1
计算 $S_i^u(t)$，$I_i^u(t)$，$D_i^u(t)$，$SID_i^u(t)$	$O(7)$	$(G+N+6)N$
S^u-S^u，I^u-I^u，S^u-I^u、I^u-S^u，S^u-D^u，I^u-D^u	$O((N+4L+6)nE_0/20)$	$(G+N+6)(N+7)$
状态保持	$O((1-7E_0/10)n)$	$(G+N+6)(N+7)$
目标函数计算	$O(n)$~$O(n^2)$	$(G+N+6)(N+7)$
生长算子	$O(3n)$	$(G+N+6)(N+7)$
结果输出	$O(n)$	1

2 H7N9 算法的参数选择

H7N9 算法参数包括两部分，一部分是 H7N9 传染病动力学模型参数，该部分参数为算法内置参数，无需用户再进行设置；另一部分是算法运行控制参数，此类参数需要用户根据情况进行设置。

(1) H7N9 传染病动力学模型参数确定方法。H7N9 传染病动力学模型参数的选择依据是确保 $S_i^1(t)$，$I_i^1(t)$，$D_i^1(t)$，$S_i^2(t)$，$I_i^2(t)$，$D_i^2(t)$ 具有足够的随机性。依据文献[14]介绍的参数取值方法并经随机化后，可得 d_1=*Rand*(0.2，0.4)，

β_1=*Rand*(0.4，0.9)，γ_1=*Rand*(0.1，0.2)，r_1=*Rand*(0.3，0.5)，d_2=*Rand*(0.2，0.4)，β_2=*Rand*(0.1，0.2)，γ_2=*Rand*(0.2，0.4)，r_2=*Ran*d(0.3，0.5)。应用此取值策略，任取 $I_i^2(t)$ 测试情况如图 4 所示。从图 4 可知，$I_i^2(t)$ 具有极好的随机性。参数 d_1，β_1，γ_1，r_1，d_2，β_2，γ_2，r_2 的取值方法可作为算法内置参数进行设置，无须用户干预。

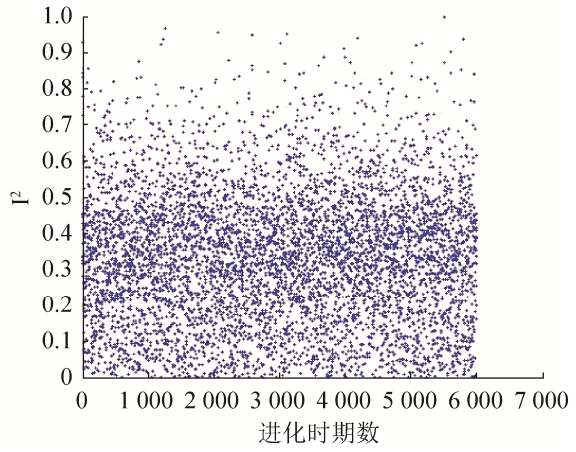


图 4 状态 I^2 出现的随机性
Fig. 4 Randomness of state I^2

(2) 算法运行控制参数设置方法。H7N9 算法的运行控制参数有：演化时期数 G、全局最优解计算误差 ε、个体特征被传染病攻击的最大概率 E_0、施加影响的个体数 L、鸡数 N_1、工人数 N_2。G 和 ε 是 2 个互补参数，只要满足 1 个即可。ε 由所求解的工程问题决定，通常可取 ε=10^{-5}~10^{-8} 即可，G 由计算设备性能决定，G 可取充分大，不妨设 G=10^8~10^{10}。H7N9 算法关键参数只有 E_0，L，N_1 和 N_2，可令 N=N_1=N_2。下面主要讨论关键参数 E_0、L，N 的取值方法。由于 BUMP 优化问题极难求解，故下面以 BUMP 优化问题为例来探明 E_0，L，N 的取值方法。BUMP 优化问题如下：

$$\begin{cases} \min Y_0(\boldsymbol{x}) = -\left|\sum_{i=1}^{n}\cos^4(x_i) - 2\prod_{i=1}^{n}\cos^2(x_i)\right| \Big/ \sqrt{\sum_{i=1}^{n} i x_i^2} \\ \text{s.t.} \quad \prod_{i=1}^{n} x_i \geqslant 0.75, \sum_{i=1}^{n} x_i \leqslant 7.5n, 0 < x_i \leqslant 10 \end{cases}$$

当 L 取不同值时，采用 H7N9 算法求解 BUMP 优化问题，令 n=50，E_0=0.01，N=200，G=10^8，运行 50 次，表 3 描述了 L 与平均最优目标函数值(*AvgO*)和平均 CPU 时间(*AvgT*)之间的关系。结果表明，当 L=3~6 时，*AvgO* 的精度达到最高，而 *AvgT* 增加较低。由此可见，L=3~6 为 L 的最佳取值区间。

表 3 L 与 *AvgO* 和 *AvgT* 之间的关系
Tab. 3 Relationship of L with *AvgO* and *AvgT*

L	*AvgO*	*AvgT*/s
1	–0.794 865 425 379 465	1 914
2	–0.817 165 234 565 718	1 923
3	**–0.828 543 760 029 943**	**2 176**
4	**–0.828 508 177 784 466**	**2 753**
5	**–0.828 417 653 467 965**	**2 476**
6	**–0.828 476 954 236 965**	**2 771**
7	–0.820 176 342 579 564	3 275
8	–0.822 968 795 346 798	3 827
9	–0.821 578 076 534 786	4 219
10	–0.823 397 653 697 658	5 524

令 n=50，N=200，L=3，G=10^8，H7N9 算法运行 50 次。表 4 描述了参数 E_0 与 *AvgO* 和 *AvgT* 之间的关系。

表 4 E_0 与 *AvgO* 和 *AvgT* 之间的关系
Tab. 4 Relationship of E_0 with *AvgO* and *AvgT*

E_0	Avg*O*	Avg*T*/s
0.001	–0.738 746 537 653 485	4 221
0.002	–0.748 885 324 875 453	3 378
0.004	–0.786 168 905 347 567	2 756
0.006	–0.806 198 954 687 665	2 084
0.008	**–0.833 787 534 785 634**	**2 010**
0.01	**–0.833 990 865 347 065**	**1 932**
0.02	**–0.833 948 734 676 534**	**1 902**
0.04	**–0.834 298 476 335 786**	**2 022**
0.06	**–0.834 848 753 467 865**	**2 123**
0.08	**–0.834 998 657 485 876**	**2 257**
0.1	**–0.834 608 745 378 546**	**2 325**
0.2	**–0.834 788 753 467 859**	**2 367**
0.4	–0.795 878 876 556 856	4 714
0.6	–0.792 388 765 847 654	5 238
0.8	–0.693 437 849 397 643	6 879
1	–0.759 189 034 659 765	9 113

表 4 结果表明，当 E_0=0.008~0.2 时，*AvgO* 的精度相对较高，且 *AvgO* 较少；当 E_0>0.2 时，*AvgT*

增加很大，且 *AvgO* 精度也大大降低；特别是当 E_0=1 时，无法获得最佳解。由此可见，E_0=0.008~0.2 为 E_0 的最佳取值区间。

令 N=200，L=3，G=10^8，H7N9 算法运行 50 次。表 5 描述了 *AvgO*，n，N 和 *AvgT* 之间的关系。

表 5 E_0，n，N 与 *AvgO* 和 *AvgT* 之间的关系
Tab. 5 Relationship of E_0, n, N with *AvgO* and *AvgT*

n	N	E_0	Avg*O*	Avg*T*/s
30	100	0.01	–0.810 987 465 764 554	772
	200	0.01	–0.809 897 845 845 334	3 102
	300	0.01	–0.811 998 457 863 478	2 087
	400	0.01	–0.814 283 254 785 345	5 679
60	100	0.01	–0.828 489 795 475 346	1 152
	200	0.01	–0.826 690 849 457 846	3 079
	300	0.01	–0.825 983 457 845 364	3 085
	400	0.01	–0.826 308 924 841 98	4 297
100	100	0.01	–0.834 887 458 746 643	2 017
	200	0.01	–0.836 526 006 633 228	2 497
	300	0.01	–0.835 589 745 784 535	5 198
300	50	0.01	–0.842 589 795 478 463	2 679
	100	0.01	–0.835 145 907 846 556	4 645
	200	0.01	–0.847 490 854 874 656	9 435
500	50	0.01	–0.828 490 845 897 453	4 129
	100	0.01	–0.836 280 745 978 425	7 621
	200	0.01	–0.827 090 845 704 263	13 715
800	50	0.001	–0.735 690 853 487 426	8 107
		0.01	–0.792 288 745 378 534	6 279
	100	0.001	–0.735 676 253 497 263	14 692
		0.01	–0.806 487 053 474 656	14 713
	200	0.001	–0.783 817 374 574 546	37 542
		0.01	–0.825 587 457 865 464	28 734
1 000	50	0.001	–0.745 979 342 578 963	9 817
		0.01	–0.749 780 342 553 426	8 347
	100	0.001	–0.738 390 342 690 764	18 429
		0.01	–0.772 270 453 783 426	20 152
	200	0.001	–0.759 078 452 879 642	31 276
		0.01	–0.807 690 845 786 346	35 729

从表 5 可以看到：

(1) 当 n 增加时，消耗的 *AvgT* 大大增加；

(2) 对于给定的 n，如果 N 增加，消耗的 *AvgT* 也大大增加，但 *AvgO* 的精度有增有减。此表明，增加 N 不一定会提升 *AvgO* 的精度；

(3) 对于给定的 n 和 N，如果 E_0 增加，*AvgO* 的精度也会增加，但消耗的 *AvgT* 可能增加或减少。

因此，如果 n≥500，N=100~200 就足够了；如果 n<500，N=200 就足够了。

3 H7N9 算法与其他算法的比较

本文使用 CEC2013[15]所提供的国际上通用的基准函数来测试 H7N9 的性能。CEC2013 包含有 28 个经过精心设计的基准测试函数，这 28 个基准测试函数共分 3 类，第 1 类是由 F1~F5 等 5 个单峰函数组成，这些单峰函数是由一些著名的极难求解函数经改造而得，它们包含有极高的条件数，主要用于测试算法的求精能力；第 2 类是由 F6~F20 等 15 个多峰函数组成，这些多峰函数也是由一些著名的极难求解函数经旋转平移后而形成，主要用于测试算法的探索能力；第 3 类是由 F21~F28 等 8 个复合函数组成，这些复合函数是由若干个第 1 类和第 2 类函数经复杂组合而形成，其函数表达式异常复杂，主要用于同时测试算法的综合能力，即求精能力和探索能力的协调性。本文选择了 6 个基准函数，每类选 2 个，如表 6 所示。

表 6 基准函数优化问题
Tab. 6 Benchmark function optimizations

基准函数		每个变量的范围	理论全局最优解	理论全局最优目标函数值
单峰函数类	F2	[–100,100]	***O***	–1 300
	F3	[–100,100]	***O***	–1 200
多峰函数类	F15	[–100,100]	未知	未知
	F19	[–100,100]	***O***	500
复合函数类	F22	[–100,100]	未知	未知
	F28	[–100,100]	未知	未知

在表 6 中，优化问题的维数为 n；***O*** 是一个 n 维决策向量，***O*** 的值随机产生。这些基准函数的形式可参见文献[15]。

用 H7N9 算法去求解表 6 所示的基准函数，H7N9 的参数是 n=50，ε=10^{-10}，N=200，L=3，E_0=0.01，G=10^{10}。选择 7 种优化算法与 H7N9 算法进行比较，这些算法包括：RC-GA[3]，DASA[16]，

NP-PSO[17]，BBO[5]，DE[6]，SaDE[19]和 ABC[18]。计算时，7 种优化算法的参数按表 7 进行初始化。

用这些算法独立求解每个基准函数 51 次，表 8 列出了平均最优目标函数值(Average)、中位数(Median)、标准偏差(STD)、最优目标函数值的最小值(Min)和最大值(Max)、平均适应度计算次数(FE），并对每种算法进行排序。表 8 的 Rank1 是按“平均最佳目标函数值的精度”进行的排名，Rank2 是按“平均最佳目标函数值的精度+平均适应度计算次数的多少”进行的排名。

表 7　7 种优化算法的参数
Tab. 7　Parameters of 7 optimization algorithms

优化算法	参数
RC-GA	染色体数目 N=100,变异率=0.01,父个体数量=0.5N,G=1.0E+10
DASA	蚂蚁的数量 m=37,**离散基数** b=10,信息素衰减率 ρ=0.2,全局规模增长因素 s_+=0.02,全局规模递减因素 s_-=0.01,变量的最大精度 ε=1.0e-15,G=1.0E+10
NP-PSO	N=100,G=1.0E+10
BBO	生境修正概率=1,基因转移概率=[0,1],概率的数值积分步长=1,每个岛屿的最大迁移和迁移率=1, 变异率=0.02,N=100,精英=2,G=1.0E+10
DE	权重因子 F=0.5,交叉常数 CR=0.9,N=100,G=1.0E+10
SaDE	加权因子区间=[0.45,0.55];交叉常数区间=[0.85,0.95],N=100,G=1.0E+10
ABC	工蜂或侦察蜂=100,测试次数=100n, G=1.0E+10

表 8　H7N9 算法和其他 7 种算法求解基准函数时所得的最优解
Tab. 8　Optimal solution of H7N9 algorithm and other 7 algorithms when solving the reference function

基准函数	H7N9	RC-GA	DSDA	NP-PSO	BBO	DE	SaDE	ABC
F2								
Average	**–1.3000E+03**	8.3225E+06	2.8240E+06	3.1023E+08	1.0861E+07	–1.2936E+03	–1.2995E+03	3.0186E+08
Median	**–1.3000E+03**	8.3225E+06	2.8240E+06	3.1023E+08	1.0861E+07	–1.2936E+03	–1.2995E+03	3.0186E+08
STD	0.0000E+00	3.3587E+02	3.4709E+02	3.3633E+02	3.6235E+02	3.0082E-03	2.6358E-03	3.1750E+02
Min	**–1.3000E+03**	8.3217E+06	2.8233E+06	3.1023E+08	1.0860E+07	–1.2936E+03	–1.2995E+03	3.0186E+08
Max	**–1.3000E+03**	8.3232E+06	2.8247E+06	3.1024E+08	1.0862E+07	–1.2936E+03	–1.2995E+03	3.0186E+08
FE	5.0292E+07	1.0139E+08	2.7997E+06	4.4721E+06	9.8624E+06	7.3458E+07	4.3885E+07	4.8080E+05
Rank1	1	5	4	8	6	3	2	7
Rank2	1	5	4	8	6	3	2	7
F3								
Average	**–1.2000E+03**	–1.1761E+03	–1.1994E+03	1.6894E+07	–1.1996E+03	**–1.2000E+03**	**–1.2000E+03**	2.3600E+04
Median	**–1.2000E+03**	–1.1761E+03	–1.1994E+03	1.6894E+07	–1.1996E+03	**–1.2000E+03**	**–1.2000E+03**	2.3600E+04
STD	0.0000E+00	2.9029E-03	3.6623E-03	3.0649E+02	3.4493E-03	0.0000E+00	0.0000E+00	3.1894E+02
Min	**–1.2000E+03**	–1.1761E+03	–1.1994E+03	1.6893E+07	–1.1997E+03	**–1.2000E+03**	**–1.2000E+03**	2.2857E+04
Max	**–1.2000E+03**	–1.1761E+03	–1.1994E+03	1.6894E+07	–1.1996E+03	**–1.2000E+03**	**–1.2000E+03**	2.4123E+04
FE	2.4861E+05	1.3597E+08	2.0624E+06	1.0646E+05	5.3366E+06	2.4995E+05	2.9842E+05	7.9065E+06
Rank1	1	6	5	8	4	1	1	7
Rank2	1	6	5	8	4	2	3	7
F15								
Average	**–3.5215E+03**	–3.5215E+03	–3.5215E+03	–3.5215E+03	–3.5215E+03	–3.5215E+03	–3.5215E+03	–3.5215E+03
Median	**–3.5215E+03**	–3.5215E+03	–3.5215E+03	–3.5215E+03	–3.5215E+03	–3.5215E+03	–3.5215E+03	–3.5215E+03
STD	7.0568E-06	7.2331E-06	3.0026E-06	2.9515E-06	3.0860E-06	2.8049E-06	8.1101E-06	3.1148E-06

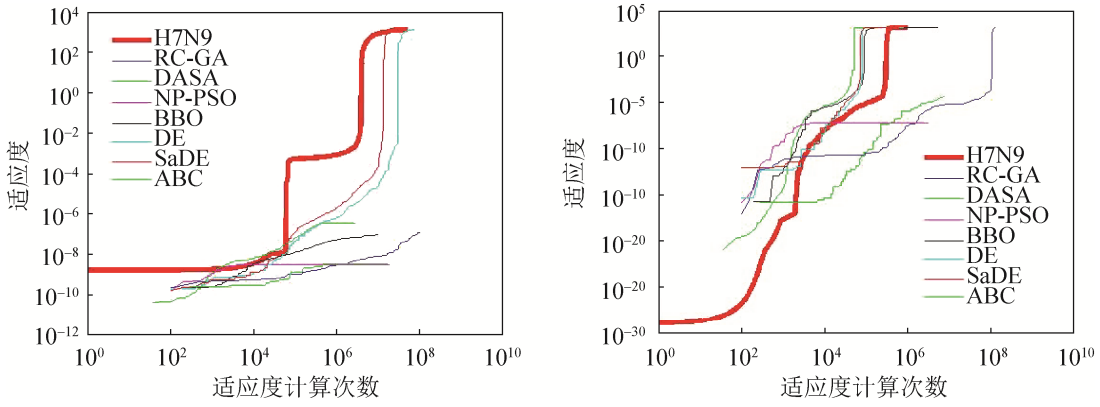
续表

基准函数	H7N9	RC-GA	DSDA	NP-PSO	BBO	DE	SaDE	ABC
Min	**–3.5215E+03**	–3.5215E+03	–3.5215E+03	–3.5215E+03	–3.5215E+03	–3.5215E+03	–3.5215E+03	–3.5215E+03
Max	**–3.5215E+03**	–3.5215E+03	–3.5215E+03	–3.5215E+03	–3.5215E+03	–3.5215E+03	–3.5215E+03	–3.5215E+03
FE	1.7195E+04	1.3721E+04	4.8194E+05	2.0287E+05	1.7373E+05	6.3975E+05	7.8498E+05	1.5443E+05
Rank1	1	2	4	7	8	5	3	6
Rank2	1	2	4	7	8	5	3	6
F19								
Average	**5.0000E+02**	9.1841E+02	5.5308E+02	1.1316E+04	5.3854E+02	5.0215E+02	5.0208E+02	5.2823E+02
Median	**5.0000E+02**	9.1841E+02	5.5308E+02	1.1319E+04	5.3854E+02	5.0215E+02	5.0208E+02	5.2823E+02
STD	**0.0000E+00**	3.3819E-03	3.1507E-03	2.8562E+02	2.8454E-03	2.9603E-03	3.2766E-03	3.1410E-03
Min	**5.0000E+02**	9.1840E+02	5.5308E+02	1.0667E+04	5.3854E+02	5.0215E+02	5.0208E+02	5.2823E+02
Max	**5.0000E+02**	9.1841E+02	5.5309E+02	1.1898E+04	5.3855E+02	5.0216E+02	5.0209E+02	5.2824E+02
FE	6.8026E+07	8.2627E+07	1.1259E+07	2.1342E+07	1.4366E+07	1.4229E+06	1.0181E+07	6.7003E+06
Rank1	1	7	6	8	5	3	2	4
Rank2	1	7	6	8	5	3	2	4
F22								
Average	**–2.5901E+03**	2.9671E+05	–2.0801E+03	–6.0283E+02	–1.5140E+02	3.9461E+03	–1.8084E+03	–1.4298E+03
Median	**–2.5901E+03**	2.9668E+05	–2.0801E+03	–6.0283E+02	–1.5140E+02	3.9461E+03	–1.8084E+03	–1.4298E+03
STD	3.1205E-05	3.3871E+02	2.9534E-05	3.1132E-05	2.9329E-05	3.3495E-02	2.6385E-05	2.9111E-05
Min	**–2.5901E+03**	2.9609E+05	–2.0801E+03	–6.0283E+02	–1.5140E+02	3.9460E+03	–1.8084E+03	–1.4298E+03
Max	**-2.5901E+03**	2.9745E+05	–2.0801E+03	–6.0283E+02	–1.5140E+02	3.9461E+03	–1.8084E+03	–1.4298E+03
FE	2.0639E+07	1.3635E+05	7.9869E+05	3.1195E+05	1.3814E+06	1.6162E+05	1.4094E+06	8.7924E+05
Rank1	1	8	2	5	6	7	3	4
Rank2	1	8	2	5	6	7	3	4
F28								
Average	**–4.7566E+03**	7.8446E+03	3.9794E+03	–2.4928E+03	4.0285E+03	3.0674E+03	3.0333E+03	4.1228E+03
Median	**–4.7566E+03**	7.8446E+03	3.9794E+03	–2.4928E+03	4.0285E+03	3.0674E+03	3.0333E+03	4.1228E+03
STD	2.8710E-02	3.2140E-02	3.2510E-02	3.0692E-05	2.6139E-02	3.0157E-02	3.4298E-02	3.1398E-02
Min	**–4.7567E+03**	7.8445E+03	3.9794E+03	–2.4928E+03	4.0284E+03	3.0673E+03	3.0333E+03	4.1227E+03
Max	**–4.7566E+03**	7.8446E+03	3.9795E+03	–2.4928E+03	4.0285E+03	3.0674E+03	3.0334E+03	4.1229E+03
FE	3.9892E+06	3.1518E+06	1.4406E+06	2.7234E+06	1.2795E+06	2.0201E+06	1.5469E+06	9.1752E+05
Rank1	1	8	5	2	6	4	3	7
Rank2	1	8	5	2	6	4	3	7
Rank1 总分	6	36	26	38	35	23	14	35
Rank2 总分	6	36	26	38	35	24	16	35
Rank1 最终排名	1	7	4	8	5	3	2	5
Rank2 最终排名	1	7	4	8	5	3	2	5

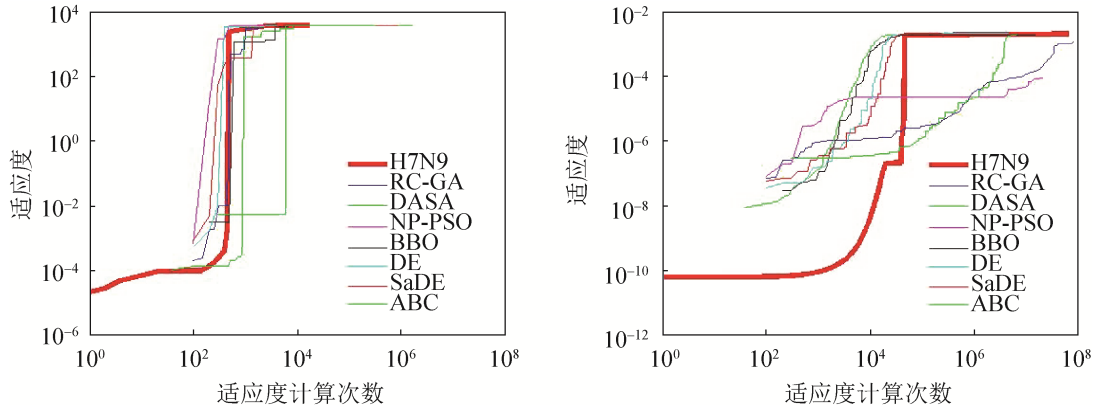
从表 8 可以看出这 8 个算法的 Rank1 和 Rank2 排名均为：

H7N9>SaDE>DE>DSDA>BBO>ABC>RC-GA>NP-PSO

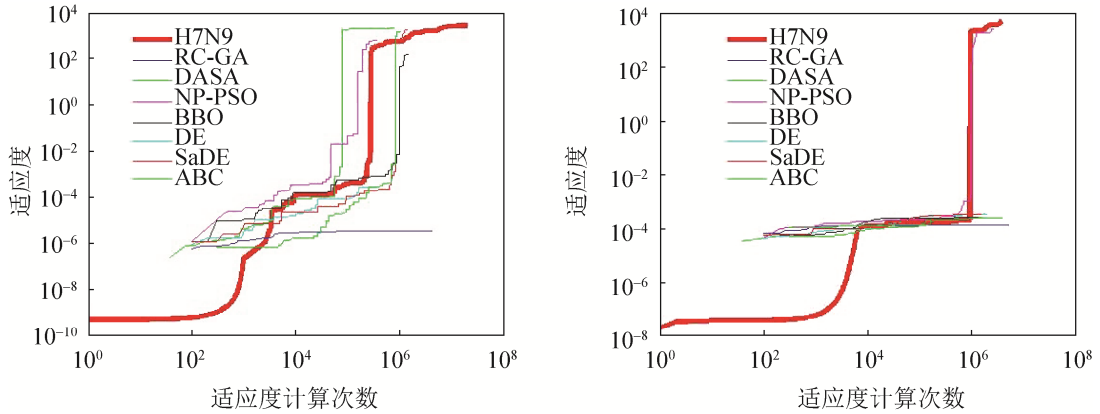
图 5(a)~(f)给出了这各算法求解 6 个基准函数时的样本收敛曲线，这些曲线的水平和垂直轴均采用对数刻度，以便突出这些曲线的变化细节。



(a) F2 (b) F3



(c) F15 (d) F19



(e) F22 (f) F28

图 5 样本收敛曲线

Fig. 5 Sample convergence curve

4 H7N9 算法的动态行为分析

4.1 算子动态行为分析

H7N9 算法包含有两类个体，每类个体包含有 6 种状态转移。每个个体的一次状态转移对应一次算子执行。图 6 描述了当 H7N9 算法求解基准函数 F3 时，其编号为 18 的鸡类个体发生 $S^1 \rightarrow S^1$，$I^1 \rightarrow I^1$，$S^1 \rightarrow I^1$，$I^1 \rightarrow S^1$，$S^1 \rightarrow D^1$，$I^1 \rightarrow D^1$ 等状态转移时，其对应的算子 S^1-S^1，I^1-I^1，S^1-I^1，I^1-S^1，S^1-D^1，I^1-D^1 被触发执行的次数与 CPU 计算时间之间的关系。从图 6 可知，每种状态转移被触发执行的次数随 CPU 计算时间随机变化，但每种状态转移的平均

https://dc-china-simulation.researchcommons.org/journal/vol32/iss5/3
DOI: 10.16182/j.issn1004731x.joss.19-0077

触发执行次数接近于水平，因此，算子 $S^1 \rightarrow S^1$、$I^1 \rightarrow I^1$，$S^1 \rightarrow I^1$，$I^1 \rightarrow S^1$，$S^1 \rightarrow D^1$，$I^1 \rightarrow D^1$ 被均匀触发执行。

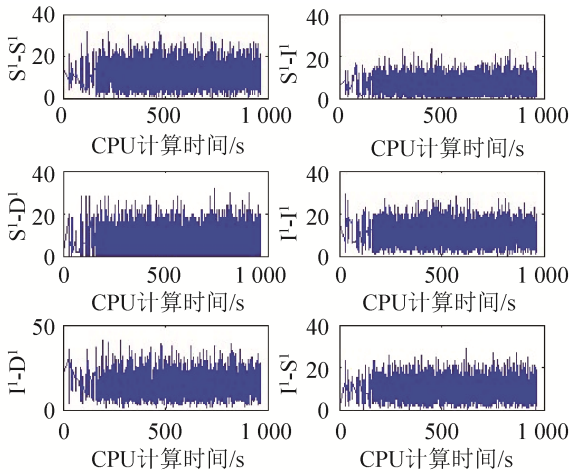


图 6 算子被触发执行的次数与 CPU 时间之间的关系
Fig. 6 Relationship between triggered number of operator is and CPU time

4.2 H7N9 算法的心率

H7N9 算法拥有两类个体，每类个体有 3 个状态；H7N9 算法在执行过程中动态和自动地将所有个体划分为 2×3=6 种状态，对应于 6 个子类。每种状态(或子类)的个体数随时间变化而变化。6 种状态下个体数量的变化可以认为是 H7N9 算法的心跳，或者称为 H7N9 算法的心律。图 7 描述了当 H7N9 算法求解基准函数 F3 时，鸡类个体处于 I^1 状态下的个体数与 CPU 时间之间的关系。

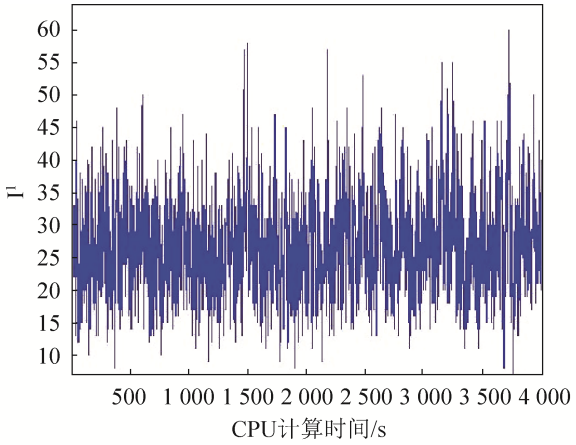


图 7 H7N9 算法的心率
Fig. 7 Heart rate of H7N9 algorithm

从图 7 可知，鸡类个体处于 I^1 状态的个体数随时间随机变化，但处于每个状态的平均个体数是稳定的，且接近于水平。

4.3 个体的动态行为分析

在 H7N9 算法中，每个个体都有让自己变得更强壮的本能，这是 H7N9 算法求解优化问题时可以收敛的基础。为了说明个体的动态行为，当我们使用 H7N9 算法求解表基准函数 F3 时，我们随机选择一个个体，如编号为 18 的工人类个体。图 8 描述了工人类个体 18 在搜索空间进行搜索时的运动轨迹。

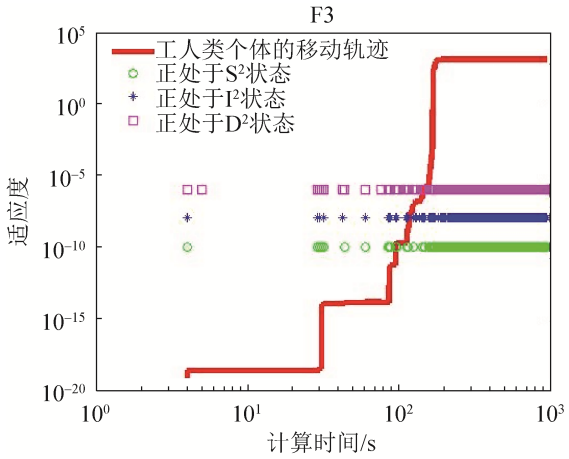


图 8 18 号工人类个体的移动轨迹及其状态转换
Fig. 8 Moving track and state transition of human being No.18

从图 8 可以看出，工人类个体 18 并没有朝着更坏的方向移动，因为该个体对应的适应度值一直在增加。

4.4 求精和探索能力及其协调性分析

(1) 求精能力分析。从表 8 可知，H7N9 算法求解 F2，F3 时，H7N9 算法均能获得其理论全局最优解，此说明 H7N9 算法较其他被比较算法具有更好的求精能力。从图 5(a)知，H7N9 算法的收敛曲线均在其他 7 个被比较算法的左侧，说明 H7N9 算法求解 F2 时要快于其他算法；从图 5(b)知，H7N9 算法求解 F3 时，其收敛曲线介于其他 7 个被比较算法之间，说明 H7N9 算法求解 F3 时要比一些算

法慢，比另外一些算法快；但对照表 8，当 H7N9 与 DE，SaDE 算法具有相同的求精能力，但 H7N9 算法的平均速度快于 DE 和 SaDE 算法。

(2) 探索能力分析。从图 5(c)和(d)可知，H7N9 算法的收敛曲线在很多时间段内容较其算法的收敛曲线陡，此说明与其他算法相比，H7N9 算法提升适应度的耗时很短，说明 H7N9 算法探索新空间的能力更强。

(3) 求精和探索能力的协调性分析。从图 5(e)和(f)可知，与其他被比较算法相比，H7N9 算法求解 F22，F28 时，其收敛曲线的缓(求精能力)和陡(探索能力)交替出现，且缓和陡的持续时间均较长，此表明 H7N9 算法的求精和探索能力的协调性均优于其他被比较算法。

5 结论

社区中爆发人感禽流感是一个复杂的自然演化场景。依据该场景，本文采用跨物种传播 H7N9 传染病模型提出了一种新的群智能优化算法，即 H7N9 算法。与其他算法相比，H7N9 算法具有如下优点：

(1) H7N9 算法中包括有形态为 S^u-S^u，I^u-I^u，S^u-I^u，I^u-S^u，S^u-D^u，I^u-D^u 的 12 个算子，拥有 2 种不同物种类型的个体，可显著地提升算法的搜索能力，从而确保该算法具有优良的性能。

(2) 在 H7N9 算法中，S^u-S^u，I^u-I^u 算子可利用强壮个体的特征来改善虚弱个体的特征，从而提升算法的求精能力；S^u-I^u，I^u-S^u 算子可改良个体的适应度分布特征，从而提升算法的探索能力；S^u-D^u，I^u-D^u 算子可使极虚弱个体得到有效清除，从而降低算法陷入局部陷阱的概率。

(3) H7N9 算法利用 H7N9 病毒只攻击个体的极少部分特征这一优势而获得每次只需要处理 n/1000~n/100 个变量这一能力，故当求解复杂优化问题，特别是高维优化问题时，能够显著提升收敛速度。

(4) H7N9 算法搜索过程具有 Markov 特性和“步步不差”特性，可确保 H7N9 算法具有全局收敛性。

该算法今后的改进方向为：

(1) 已经发现，某些传染病能够跨不低于 3 个物种，可以利用 H7N9 算法的设计思路，提出跨多物种的传染病优化算法。

(2) 将 H7N9 算法的状态数从当前的 S(易感)、I(染病)、D(死亡)等 3 个状态扩展到 S(易感)、E(暴露)、I(发病)、V(免疫)、R(治愈)、D(死亡)等 6 个状态，从而使 H7N9 算法拥有更多的算子。

(3) 深入分析 S^u-S^u，I^u-I^u，S^u-I^u，I^u-S^u，S^u-D^u，I^u-D^u 的性能。

(4) H7N9 算法依赖于人感禽流感传染病的研究成果，将其最新研究成果纳入到 H7N9 算法的算子研究中，是本算法下一步的重点研究方向。

参考文献：

[1] Bazaraa M S, Sherali H D, Shetty C M. Nonlinear Programming—Theory and Algorithms[M]. New York: John Wiley & Sons, 1993.

[2] Holland J H. Adaptation in Natural and Artificial Systems[M]. Ann Arbor: University of Michigan Press, 1975.

[3] Chuang Y C, Chen C T, Hwang C. A simple and efficient real-coded genetic algorithm for constrained optimization[J]. Applied Soft Computing (S1568-4946), 2016, 38(1): 87-105.

[4] Serani A, Leotardi C, Iemma U, et al. Parameter selection in synchronous and asynchronous deterministic particle swarm optimization for ship hydrodynamics problems[J]. Applied Soft Computing (S1568-4946), 2016, 49(2): 313-334.

[5] Al-Roomi A R, El-Hawary M E. Metropolis biogeography-based optimization[J]. Information Sciences (S0020-0255), 2016, 360(5): 73-95.

[6] Mukherjee R, Debchoudhury S, Das S. Modified differential evolution with locality induced genetic operators for dynamic optimization[J]. European Journal of Operational Research (S0377-2217), 2016, 253(1): 337-355.

[7] Souza S S F, Romero R, Pereira J, et al. Artificial immune algorithm applied to distribution system reconfiguration with variable demand[J]. International

Journal of Electrical Power and Energy Systems (S0142-0615), 2016, 82(5): 561-568.

[8] Huang G Q. SIS epidemic model-based optimization[J]. Journal of Computational Science (S1877-7503), 2014, 5: 32-50.

[9] Liu Y, Wu X Y, Li P, et al. Research progress of ecological security assessment based on the risk of avian influenza[J]. Acta Ecologica Sinica (S1872-2032), 2018, 38(14): 5255-5269.

[10] 张斯钰, 黄一伟, 胡世雄, 等. 湖南省 2005-2017 年人感染禽流感流行病学特征分析[J]. 中华疾病控制杂志, 2018, 22(10): 1037-1040.
Zhang Siyu, Huang Yiwei, Hu Shixiong, et al. Epidemiological characteristics of human avian influenza in Hunan Province from 2005 to 2017[J]. Chinese Journal of Disease Control, 2018, 22(10): 1037-1040.

[11] 董泽丰, 夏瑜, 王笛, 等. 苏州市人感染 H7N9 禽流感聚集性疫情调查分析[J]. 检验医学与临床, 2018, 15(21): 3254-3256.
Dong Zefeng, Xia Yu, Wang Di, et al. Investigation and analysis of human infection with H7N9 avian influenza in Suzhou [J]. Laboratory Medicine and Clinic, 2018, 15(21): 3254-3256.

[12] Kermack W O, Mckendrick A G. Contributions to the mathematical theory of epidemics[C]. Proceedings of the Royal Society of London. London: the Royal Society of London, 1927, A115: 700-721.

[13] Kermack W O, Mckendrick A G. Contributions to the mathematical theory of epidemics[C]. Proceedings of the Royal Society of London. London: the Royal Society of London, 1932, A138: 55-83.

[14] 杨伟. 传染病动力学的一些数学模型及其分析[D]. 上海: 复旦大学, 2010.
Yang Wei. Some Mathematical Models of Infectious Disease Dynamics and Their Analysis[D]. Shanghai: Fudan University, 2010.

[15] Liang J J, Qu B Y, Suganthan P N, et al. Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter rptimization[R]. Singapore: Nanyang Technological University, 2013.

[16] Korošec P, Šilc J, Filipic B. The differential ant-stigmergy algorithm[J]. Information Sciences (S0020- 0255), 2012, 192(5): 82-97.

[17] Beheshti Z, Shamsuddin S M. Non-parametric particle swarm optimization for global optimization[J]. Applied Soft Computing (S1568-4946), 2015, 28(5): 345-359.

[18] Li G H, Cui L Z, Fu X H, et al. Artificial bee colony algorithm with gene recombination for numerical function optimization[J]. Applied Soft Computing (S1568-4946), 2017, 52(7): 146-159.

[19] Zhao Z W, Yang J M, Hu Z Y, et al. A differential evolution algorithm with self-adaptive strategy and control parameters based on symmetric Latin hypercube design for unconstrained optimization problems[J]. European Journal of Operational Research (S0377-2217), 2016, 250(1): 30-45.