

# 求解背包问题的病毒协同进化粒子群算法

高 芳, 崔 刚, 吴智博, 刘宏伟, 杨孝宗

(哈尔滨工业大学 计算机科学与技术学院, 哈尔滨 150001, gaofang@hit.edu.cn)

**摘 要:** 为提高粒子群算法的搜索性能, 提出一种基于病毒进化理论的改进离散粒子群算法: 病毒协同进化粒子群算法。在粒子群中引入生物病毒机制和宿主与病毒基于感染操作的思想, 病毒采用与粒子等长的编码方式, 执行反向代换、结合等操作, 利用病毒的水平感染和垂直传播能力较好地维持个体的多样性和对解空间的局部搜索能力。通过解决背包问题对算法进行验证, 仿真表明所提算法搜索性能优于遗传算法、模拟退火及标准粒子群等其他算法。该算法能有效求解背包问题等 NP 难题。

**关键词:** 粒子群算法; 病毒; 背包问题; 病毒感染

**中图分类号:** TP301.6; TP18 **文献标识码:** A **文章编号:** 0367-6234(2009)06-0103-05

## Virus-evolutionary particle swarm optimization algorithm for knapsack problem

GAO Fang, CUI Gang, WU Zhi-bo, LIU Hong-wei, YANG Xiao-zong

(School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China, gaofang@hit.edu.cn)

**Abstract:** To improve the search capability of particle swarm algorithm, an improved discrete particle swarm optimization algorithm based on virus evolution theory is proposed and named as virus-evolutionary discrete particle swarm optimization (VEPSO) algorithm. Biological virus mechanism and the infection-based operation between host and virus are introduced in the particle swarm. Virus individual is coded with the same length as particle, and it executes infection and incorporation operations. The horizontal infection and vertical propagation of virus are fully used to maintain the individual diversity and local search capability in solution space. This algorithm is verified by solving knapsack problem. Simulation results show that the search capability of VEPSO algorithm is better than that of genetic algorithm, simulated annealing and standard PSO algorithm. This algorithm is able to effectively solve knapsack and other NP-hard problems.

**Key words:** particle swarm algorithm; virus; knapsack problem; virus infection

粒子群算法是 Kennedy 和 Eberhart 提出的一种群体智能优化方法<sup>[1]</sup>, 是解决优化问题的有效工具, 该算法在很多实际应用中都取得了成功<sup>[2~4]</sup>。但是粒子群算法存在易于陷入局部最优及搜索精度差的缺陷, 严重影响了该算法的性能。

防止粒子陷入局部最优以及陷入局部最优后如何逃离, 以及提高算法的搜索精度是改进粒子群算法的直接出发点。本文从生物病毒系统中获得启发来改进粒子群算法, 生物病毒系统作为免疫系统的一个主要组成部分, 它所具有的许多信息处理机制和功能特点, 如病毒个体和宿主个体的自然划分, 不同个体的不同行为, 病毒与宿主基于感染操作的协同等<sup>[5,6]</sup>, 对于改进进化算法, 开发设计新的计算智能模型具有重要的意义。

以标准离散粒子群算法为基础, 从生物病毒机制中抽取出适合改进粒子群算法的一些特征,

收稿日期: 2008-03-20。

基金项目: 中国高技术研究发展计划重大资助项目 (2006AA01A103)。

作者简介: 高 芳 (1973—), 女, 博士;

崔 刚 (1947—), 男, 教授, 博士生导师;

吴智博 (1954—), 男, 教授, 博士生导师;

杨孝宗 (1939—), 男, 教授, 博士生导师。

提出一种病毒协同进化粒子群算法(VEPSO).在粒子群中引入若干病毒个体,通过执行反向代换、结合等操作来模拟生物病毒的感染、复制过程,利用病毒的水平感染和垂直传播能力维持粒子个体的多样性和提高解空间的局部搜索能力.

## 1 背包问题描述

背包问题的一般提法为:已知  $n$  个物品  $s_1, s_2, \dots, s_n$  的重量及其价值分别为  $w_j > 0$  和  $c_j > 0 (j = 1, 2, \dots, n)$ , 假设背包的容量为  $V > 0$ , 引入变量  $x_j$  为

$$x_j = \begin{cases} 1 & \text{物品 } s_j \text{ 放入背包} \\ 0 & \text{否则} \end{cases}, j = 1, 2, \dots, n. \quad (1)$$

则该问题的数学模型为

$$\max \left( \sum_{j=1}^n c_j x_j \right), \quad (2)$$

$$\text{s. t. } \begin{cases} \sum_{j=1}^n w_j x_j \leq V \\ x_j \in (0, 1), j = 1, 2, \dots, n \end{cases}. \quad (3)$$

## 2 离散粒子群算法

在离散粒子群算法中,粒子在状态空间的取值和变化只限于 0 和 1 两个值,而速度的每一位  $v_{ij}$  代表位置每一位  $x_{ij}$  取值为 1 的可能性<sup>[7]</sup>. 因此在离散粒子群中  $v_{ij}$  的更新公式依然保持不变,但是  $pbest_{ij}$  和  $gbest_j$  只在  $\{0, 1\}$  内取值,而位置更新为

$$\text{if } r < s(v_{i,j}) \text{ then } x_{i,j} = 1, \text{ else } x_{i,j} = 0. \quad (4)$$

其中,  $r$  为从  $U(0, 1)$  分布中产生的随机数. 函数  $s(v_{i,j})$  为一个 sigmoid 函数,能把  $v_{i,j}$  限制在  $[0, 1]$  之间的逻辑映射,随着  $s(v_{i,j})$  接近 0, 粒子的位置就可能固定在 0 值,会有更少的改变机会. 与连续粒子群算法不同的是,粒子实际的位置没被调整. 另外为了防止  $v_{i,j}$  饱和,可以将其钳位在  $[-4.0, 4.0]$  之间.

实验结果表明,对于大多数测试函数,离散粒子群算法都比遗传算法速度快,尤其在问题的维数增加时更为明显,这扩展了粒子群算法在组合优化问题的应用.

## 3 病毒协同进化粒子群算法

根据病毒进化理论,病毒是一种特有的生物,具有一种特有的感染功能. 它能获得一个个体的染色体基因,并且感染给另一个个体,使得该个体

的部分染色体发生相应的变化,从而改变该个体的遗传信息. 这种遗传信息又通过遗传传递给下一代,大大加速了生物的进化速度.

基于上述思想,VEPSO 算法产生两种群体:粒子群主群体和病毒群体. 主群体对应问题的解空间,执行式(4)的操作,实施解空间的全局搜索. 病毒群体进行病毒感染操作,在同代个体之间横向传递进化基因,实施解空间的局部搜索. VEPSO 算法将主群体的全局进化和病毒群体的局部进化进行动态结合,从而快速得到问题的全局近似最优解.

### 3.1 粒子和病毒个体编码及适应度

主群体粒子编码方式为:  $X_i^k$  为粒子群中的第  $i$  个粒子在进化的第  $k$  代的位置值. 每一个  $X_i^k$  为背包问题的一个解,其中,  $X_i^k = [x_{i1}^k, x_{i2}^k, \dots, x_{in}^k]$ ,  $n$  为粒子群中粒子具有的维数,在背包问题中为有  $n$  个物品等待被选择是否装入背包. 如果  $x_{ij}^k = 0$  为第  $i$  个粒子在进化的第  $k$  代时不将第  $j$  个物品装入背包,否则将第  $j$  个物品装入背包.  $V_i^k$  为粒子群中的第  $i$  个粒子在进化的第  $k$  代的速度值. 其中,  $V_i^k = [v_{i1}^k, v_{i2}^k, \dots, v_{in}^k]$ .

病毒个体产生于主个体,是主个体的子串,一种是位长与主个体不等的编码方案,如 10 和 001 等. 这里,采用另外一种位长和主个体相等的编码方案,串中包含通配符,串中除通配符以外的字符,称为有效字符,如 10 \* 1 \* \* 1 是产生于主个体 1011001 的病毒个体.

一个病毒个体可以感染若干个主个体,每个病毒个体都有一个适应度函数  $fitvirus_i$ . 病毒个体的适应度函数是用它所感染的主个体感染前后的适应度函数的变化来表示. 假设病毒  $i$  所感染的主个体的集合为  $U$ ,  $U$  中主个体  $j$  被感染前后的适应度函数分别表示为  $fitparticle_j$  和  $fitparticle'_j$ , 则病毒  $i$  的适应度函数为

$$fitvirus_{i,j} = fitparticle'_j - fitparticle_j, \quad (5)$$

$$fitvirus_i = \sum_{j \in S} fitvirus_{i,j}. \quad (6)$$

### 3.2 病毒个体的感染操作

病毒的感染操作包括:

1) 反向代换 (也称为病毒感染), 用病毒染色体基因替换主染色体中相应基因, 从而产生新的子个体, 如图 1 所示. 每个病毒个体以概率  $infrate_i$  感染每个主个体, 如果被感染的主个体的适应度减少了, 则用其父个体替换该子个体.

2) 结合 (incorporation) 从个体中析取子串, 作为新的病毒个体. 结合操作分为两种:

① 复制(copy). 随机选择一个主个体,主染色体每个基因以概率  $P_i$  代换病毒染色体中相应基因,从而产生有效字符更多的新病毒个体. 初始病毒个体也是通过复制产生,即通过一个主染色体,以概率  $P_i$  复制到一个无感染能力(infectless)的病毒个体上,如图2所示.

② 删减(cut). 病毒染色体每个基因以概率  $P_j$  用通配符代换,如图3所示.

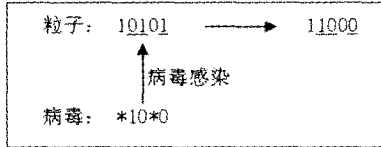


图1 病毒感染操作

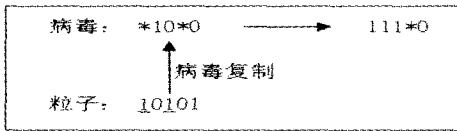


图2 复制

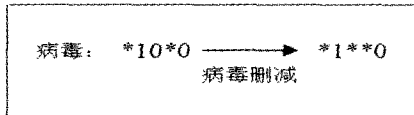


图3 删减

在病毒感染操作之后,每个病毒按照概率  $fit_{virus_i}$  进行结合进化,如果  $fit_{virus_i} > 0$ ,病毒个体从感染的粒子中析取新的子串来完成自己的进化. 如果  $fit_{virus_i} < 0$ ,病毒将删除自身的一些字符串. 事实上,病毒个体长度的增加或删除依赖于结合概率  $P_i$ . 对于每个病毒而言,都有一个病毒感染生命力函数  $infrate_i$ .  $infrate_i$  定义为

$$infrate_{i,t+1} = \begin{cases} (1 + \alpha) \cdot infrate_{i,t} & \text{if } fit_{virus} > 0 \\ (1 - \alpha) \cdot infrate_{i,t} & \text{otherwise} \end{cases} \quad (7)$$

其中,  $\alpha$  为系数( $> 0$ ). 如果  $fit_{virus_i}$  值增大,则  $infrate_i$  也增加. 随着  $infrate_i$  值的增大,病毒感染力也随之增大. 因此,每个病毒的生命力定义为

$$life_{i,t+1} = r \times life_{i,t} + fit_{virus_i} \quad (8)$$

其中,  $i$  为粒子进化的代数,  $r$  为病毒的生命力衰减率. 如果  $r$  为负数,则病毒粒子通过结合操作从随机选取的粒子中析取新的子串来更新自己.

## 4 算法描述

病毒协同进化粒子群算法描述为:

1) 输入算法选取的各种参数值.

2) 初始化粒子群:随机的产生  $N$  个粒子的位置  $X_1, X_2, \dots, X_N$ , 及粒子的速度  $v_{ij}$ ,  $i = 1, 2, \dots, N$ , 和  $j = 1, 2, \dots, n$ , 其中,  $v_{ij} \sim U(0, 1)$ .

3) 初始化病毒粒子群:随机产生  $0.2 \times N$  病毒粒子.

4) 按照具体问题的目标函数公式评价每个粒子.

5) 按照病毒  $i$  的适应度函数评价每个病毒.

6) 更新粒子个体及全局的最好位置. 如果  $f(pbest_i) < f(X_i)$ , 则  $pbest_i = X_i$ , 并在  $f(pbest_i)$  中搜索  $f_{max}$ . 如果  $f(gbest) < f_{max}$ , 则  $gbest = X_{max}$ ,  $X_{max}$  是与  $f_{max}$  相关的粒子.

7) 速度更新. 按速度更新公式在  $v_{max}$  和  $v_{min}$  范围内更新粒子的速度.

8) 位置更新. 按照位置更新公式更新第  $i$  个粒子的位置.

9) 病毒感染的生命力计算,病毒感染操作.

10) 病毒生命力计算,病毒个体更新.

11) 重复3)~10)直到达到给定的最大迭代次数.

12) 输出问题的最优解.

## 5 仿真实验

为了更好地说明算法的运行效果,用 Matlab7.0 为算法编写了程序,选取文献[8~10]中的仿真实例进行求解,具体为:

仿真实例1的原始数据为:

$C = [72; 490; 651; 833; 883; 489; 359; 337; 267; 441; 70; 934; 467; 661; 220; 329; 440; 774; 595; 98; 424; 37; 807; 320; 501; 309; 834; 851; 34; 459; 111; 253; 159; 858; 793; 145; 651; 856; 400; 285; 405; 95; 391; 19; 96; 273; 152; 473; 448; 231];$

$W = [438; 754; 699; 587; 789; 912; 819; 347; 511; 287; 541; 784; 676; 198; 572; 914; 988; 4; 355; 569; 144; 272; 531; 556; 741; 489; 321; 84; 194; 483; 205; 607; 399; 747; 118; 651; 806; 9; 607; 121; 370; 999; 494; 743; 967; 718; 397; 589; 193; 369];$

$V = 11258;$

仿真实例2的原始数据为:

$C = [220; 208; 198; 192; 180; 180; 165; 162; 160; 158; 155; 130; 125; 122;$

120; 118; 115; 110; 105; 101; 100;  
100; 98; 96; 95; 90; 88; 82; 80; 77;  
75; 73; 72; 70; 69; 66; 65; 63; 60;  
58; 56; 50; 30; 20; 15; 10; 8; 5; 3;  
1];

$W = [80; 82; 85; 70; 72; 70; 66; 50; 55;$   
 $25; 50; 55; 40; 48; 50; 32; 22; 60;$   
 $30; 32; 40; 38; 35; 32; 25; 28; 30;$   
 $22; 50; 30; 45; 30; 60; 50; 20; 65;$   
 $20; 25; 30; 10; 20; 25; 15; 10; 10;$   
 $10; 4; 4; 2; 1];$

$V = 1000;$

实例仿真参数选取为:

1)一般粒子群参数选取. 罚函数因子  $Q = 100000$ ; 参数维数  $dSize = 50$  种群大小为 500, 惯性因子  $w = 1, c_1 = c_2 = 2, v_{\max} = 4, v_{\min} = -4$ , 迭代次数 200 次.

2)病毒粒子群参数选取. 病毒群体规模为群体规模的 10%, 即 50 个病毒个体, 病毒个体生命力递减率  $r = 0.9, P_i = 0.2, P_j = 0.15, \alpha = 0.1$ , 病毒感染概率:  $infrate_i = 0.02$ , 其它参数与一般粒子群相同.

不同算法求解实例 1 的仿真结果如表 1 所示, 实例 2 的仿真结果如表 2 所示. 表 1 中的部分数据来源于文献[8], 由表 1 可知, 在五种不同的算法中, 标准粒子群算法的性能最差, 最优值为 14863, 简单遗传算法和模拟退火算法性能依次递增, 病毒粒子群和遗传退火进化算法(GAEA)算法最好, 都求得了问题的最优解:16102. 明显优于文献[8]提到的遗传(GA)算法的最优值 14865, 模拟退火(SA)算法  $t = 10$ , 最优值 15844, 而 SA 算法在  $t = 100$  时的最优值 15955. 标准粒子群及病毒粒子群算法求解仿真实例 1 的迭代过程如图 4 所示.

表 1 仿真实例 1 的背包问题最好求解结果之比较

项目	最优个体 $x_i (i = 1, 2, \dots, 50)$	最优值(总价值/总重量)
简单遗传算法	101110000001111001101010	14865
	10111110111011010100000011	
模拟退火算法	001110010101110001101010	15955
	1111110001101101101000011	
遗传退火进化算法	0111100101011100011010101	16102
	0110100011011011010000110	
标准粒子群算法	1001101011010100011011100	14863
	011011101101 1111010000011	
病毒粒子群算法	0111100101011100011010101	16102
	0110100011011011010000110	

表 2 仿真实例 2 的背包问题最好求解结果之比较

项目	决策变量 $x_i (i = 1, 2, \dots, 50)$	求解结果(总价值/总重量)
简单遗传算法	1101101111101001101111111	3077/999
	1010000001010011000000011	
混合遗传算法	1101010111101101101101111	3103/1000
	1110100001010011000001000	
贪婪算法	1111111101101001101100111	3036/999
	1010000001010010000000001	
标准粒子群算法	1010110011101001110110101	2825/1000
	1110101101000011001101011	
病毒粒子群算法(病毒个体为群体规模的 20%)	1101010111101101101101111	3097/1000
	1110100001010011000001000	
病毒粒子群算法(病毒个体为群体规模的 10%)	1101010111101101101101111	3103/1000
	1110100001010011000001000	

表2中部分数据来源于文献[9],由表2可知在六种算法中,标准粒子群算法的性能最差,最好值为:2825/1000,简单遗传算法和贪婪算法性能依次递增,病毒粒子群算法和混合遗传算法性能最好,病毒粒子群试验最好值为:3097/1000.明显优于简单遗传算法的最优值 3077/999,贪婪算法的最优值 3036/999,与混合遗传算法的最优值 3103/1000 相同.

由表2还可知,病毒个体的数量对算法性能也有一定的影响,本例中病毒个体数量取值为群体规模的10%的算法结果好于取值为20%的算法.标准粒子群及病毒粒子群算法求解仿真实例2的迭代过程如图5所示.

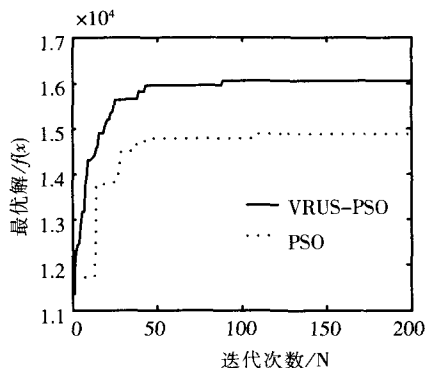


图4 实例1迭代过程

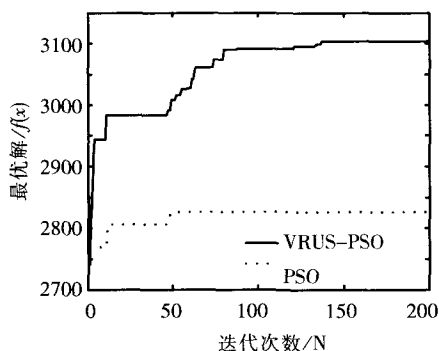


图5 实例2迭代过程

## 6 结 论

1)提出的病毒协同进化粒子群算法,充分利用了病毒所具有的水平感染和垂传播播两大特性,有效改善了粒子群算法的全局搜索能力和局部搜索能力,提高了算法的搜索精度.

2)病毒采用与粒子等长的编码方式,增加病毒感染时与主群体的亲和性以及感染的基因跨度,提高了算法的效率.

3)通过解决 NP - 难题背包问题的两个具体实例对该算法进行验证,仿真结果表明,病毒协同进化粒子群算法在维持粒子个体的多样性、保留最优搜索信息和提高算法精度等方面都得到很好的效果,明显优于其它的几种方法,所求得问题解的精确度具有很好的性能.

## 参考文献:

- [1] EBERHART R C, KENNEDY J. A new optimizer using particle swarm theory [C]//Proceedings of the Sixth International Symposium on Micro Machine and Human Science. Nagoya: IEEE, 1995: 39 - 43.
- [2] CAGNINA L, ESQUIVEL S, GALLARD R. Particle swarm optimization for sequencing problems: a case study [C]//Proceedings of the 2004 Congress on Evolutionary Computation. Nagoya: IEEE, 2004: 536 - 541.
- [3] 赵强,丁柏群. 用改进的粒子群算法求解六自由度运动平台的最大误差[J]. 机械设计, 2007, 24(6): 39 - 42.
- [4] 王灵,俞金寿. 混沌耗散离散粒子群算法及其在故障诊断中的应用[J]. 控制与决策, 2007, 22(10): 1197 - 1200.
- [5] NAOYUKI K, KOJI S, FUKUDA T. The role of virus infection in virus-evolutionary genetic algorithm [C]//Proceeding of 1996 International Conference on Evolutionary Computation. Nagoya: IEEE, 1996: 182 - 187.
- [6] NAOYUKI K, FUKUDA T. Schema representation in virus-evolutionary genetic algorithm for knapsack problem [C]//Proceeding of 1998 IEEE International Conference on Evolutionary Computation. Anchorage: IEEE, 1998: 834 - 839.
- [7] KENNEDY J, EBERHART R C. A discrete binary version of the particle swarm algorithm [C]//Proceeding of 1997 International Conference on Systems, Man and Cybernetics. Orlando: IEEE, 1997: 4104 - 4108.
- [8] 金慧敏,马良. 遗传退火进化算法在背包问题中的应用[J]. 上海理工大学学报, 2004, 26(6): 561 - 564.
- [9] 李娟,方平,周明. 一种求解背包问题的混合遗传算法[J]. 南昌航空下业学院学报, 1998, 12(3): 31 - 35.
- [10] 马慧民,叶春明,张爽. 二进制改进粒子群算法在背包问题中的应用[J]. 上海理工大学学报, 2006, 28(1): 31 - 34.

(编辑 张 红)