第 36 卷 第 8 期
2014 年 8 月

系统工程与电子技术
Systems Engineering and Electronics

Vol. 36 No. 8
August 2014

文章编号:1001-506X(2014)08-1660-08

网址:www. sys-ele. com

求解 0-1 背包问题的二进制狼群算法

吴虎胜[1,2],张凤鸣[1],战仁军[2],汪 送[2],张 超[1]

(1. 空军工程大学装备管理与安全工程学院,陕西 西安 710051;
2. 武警工程大学装备工程学院,陕西 西安 710086)

摘 要:狼群算法(wolf pack algorithm, WPA)源于狼群在捕食及其猎物分配中所体现的群体智能,已被成功应用于复杂函数求解。在此基础上,通过定义运动算子,对人工狼位置、步长和智能行为重新进行二进制编码设计,提出了一种解决离散空间组合优化问题的二进制狼群算法(binary wolf pack algorithm, BWPA)。该算法保留了狼群算法基于职责分工的协作式搜索特性,选取离散空间的经典问题——0-1 背包问题进行仿真实验,具体通过 10 组经典的背包问题算例和 BWPA 算法与经典的二进制粒子群算法、贪婪遗传算法、量子遗传算法在求解 3 组高维背包问题时的对比计算,例证了算法具有相对更好的稳定性和全局寻优能力。

关键词:进化计算;群体智能;二进制狼群算法;组合优化;0-1 背包问题

中图分类号:TP 18;TP 301.6 **文献标志码**:A **DOI**:10.3969/j.issn.1001-506X.2014.08.34

A binary wolf pack algorithm for solving 0-1 knapsack problem

WU Hu-sheng[1,2], ZHANG Feng-ming[1], ZHAN Ren-jun[2], WANG Song[2], ZHANG Chao[1]

(1. *Materiel Management and Safety Engineering College, Air Force Engineering University, Xi'an 710051, China*;
2. *Materiel Engineering College, Armed Police Force Engineering University, Xi'an 710086, China*)

Abstract: The wolf pack algorithm (WPA), inspired by swarm intelligence of wolf pack in their prey hunting behaviors and distribution mode, has been proposed and successfully applied in complex function optimization problems. Based on the designing of the move operator, the artificial wolves' position, step-length and intelligent behaviors are redesigned by binary coding, and a binary wolf pack algorithm (BWPA) is proposed to solve combinatorial optimization problems in discrete spaces. BWPA preserves the feature of cooperative searching based on job distribution of the wolf pack and is applied to 10 classic 0-1 knapsack problems. Moreover, the 3 high-dimensional 0-1 knapsack problems are tested. All results show that BWPA has better global convergence and computational robustness and outperforms the binary particle swarm optimization algorithm, the greedy genetic algorithm and the quantum genetic algorithm, especially for high-dimensional knapsack problems.

Keywords: evolutionary computation; swarm intelligence; binary wolf pack algorithm; combinatorial optimization; 0-1 knapsack problem

0 引 言

狼,被誉为"草原上的精灵",历经千百年的进化和繁衍,表现出令人叹为观止的生物集群智能,学者们也从狼群群体生存智慧中获得启示并应用于各种复杂问题的求解[1-3]。狼群算法(wolf pack algorithm, WPA)就是一种模拟狼群分工协作式捕猎行为及其猎物分配方式的群体智能算法,具有较好的计算鲁棒性和全局搜索能力,已成功应用于多个复杂函数寻优问题,尤其对于高维、多峰的复杂函数寻优效果较好[4]。而实际中,依据解空间的不同,优化问题大体可分为连续空间优化问题和离散空间优化问题,复杂函数寻优问题属于前者。因此,有必要进一步研究狼群算法的离散版本,以用于解决诸如 0-1 背包问题、投资组合和车间作业调度等问题。同时,连续空间和离散空间也可通过特定的编码进行相互转换[5]。如果能将 WPA 算法引入到离散空间中,就能大大地扩展其使用范围。本文在继承 WPA 算法基于职责分工的协作式寻优模式的基础上,引入二进制编码,提出一种二进制狼群算法用于求解经典的组合优化问题——0-1 背包问题,并通过大量的仿真对比实验验证了算法的有效性。

1 0-1 背包问题描述

0-1 背包问题是经典的组合优化问题,问题旨在寻求背

收稿日期:2013-12-02; 修回日期:2014-03-12; 网络优先出版日期:2014-03-27。
网络优先出版地址:http://www.cnki.net/kcms/detail/11.2422.TN.20140327.1428.023.html
基金项目:国家自然科学基金(71171199)资助课题

包容积限制下具有最大价值的物品装载方案。如任务调度、投资项目组合、资源分配、材料切割等很多现实问题都可抽象为 0-1 背包问题，应用非常广泛[6]。具体描述为：给定容量为 C 的背包，有 m 个待装物品，第 i 个物品的体积为 w_i，价值为 p_i，问应选择装入哪些物品才可使背包中的物品价值最大。若定义 $x_i=1$ 表示物品 i 装入包中，$x_i=0$ 表示不装入，则问题可表示为

$$\begin{cases}\max f(x)=\sum_{i=1}^{m}p_i x_i \\ \text{s.t.}\ \sum_{i=1}^{m}w_i x_i\leqslant C,\ x_i=0,1\end{cases} \tag{1}$$

0-1 背包问题是典型的描述简明但实际求解困难的问题，已被证明属于 NP-hard 问题，枚举搜索可能遍历问题的所有 2^n 个解空间[7]。诸如动态规划法、贪心算法和分支界定法等传统算法无法解决因物品种类增大而产生的“组合爆炸”问题[8]。因此，二进制粒子群算法（binary particle swarm optimization，BPSO）、二进制入侵杂草算法、二进制和声搜索算法、遗传算法、人工蜂群算法等智能算法被用于解决背包问题[9-11]。这些算法在一定程度上克服了传统算法的不足，提高了求解速度，但也并不完美：如遗传算法易陷入局部最优且局部寻优能力不足；而诸如粒子群、人工蜂群等算法的提出主要针对于连续空间优化问题，对于离散空间的优化问题求解优势并不明显[12]。

0-1 背包问题属于带约束的离散优化问题，当用二进制狼群算法（binary wolf pack algorithm，BWPA）求解时需对模型进行转化。这里引入惩罚系数对不可行解施加惩罚，最终转化为如下式所示的无约束优化问题：

$$\max F(x)=f(x)-\lambda\cdot\max\left(0,\left(\sum_{i=1}^{m}w_i x_i-C\right)\right) \tag{2}$$

式中，λ 为惩罚系数，为一足够大的数，以保证可行解的最差值都要优于不可行解的最优值。

2　二进制狼群算法

2.1　狼群系统分析

狼是群居动物，独狼很难长时间存活。狼群中头狼、探狼和猛狼在捕猎时承担着不同的责任，它们相互协作、分享信息，只求尽快搜寻并捕获猎物。头狼始终是狼群中体质最好、最凶猛和最具智慧的领头狼。它负责指挥整个狼群，其他的狼都要直接或间接地将自己所掌握的信息向其汇报，头狼据此并结合自身经验做出决策，指挥狼群的捕猎行为。探狼是狼群中的精锐，负责在狼群的领地内搜寻猎物。狼具有敏锐的嗅觉，探狼就是通过感知空气中猎物所留下的气味来判断猎物的大致方向，并始终向着气味最浓的方向搜寻。一旦探狼发现猎物的踪迹就立即向头狼报告，头狼根据具体情况进行决策，如猎物庞大，头狼就通过嚎叫的方式召唤周围的狼来捕杀猎物，听到召唤的狼则会向着猎物所在位置快速奔袭。当狼群包围了猎物以后，头狼下达进攻指令，所有的狼便会从不同的方向、不同的部位一起围攻猎物直至猎物被捕杀。而后，捕获的猎物被按照“由强到弱”的方式分配，这样一方面使得有能力搜寻和捕获猎物的强狼保持充沛的体力；另一方面，弱狼和病狼由于得不到充足的食物可能会被饿死。如此强者生存、弱者淘汰的机制，使得狼群保持着较强的群体生存活力[4]。

基于对狼群的系统分析，下文将对上述狼群捕猎行为和生存机制进行数学抽象和具体描述。

2.2　一些定义

不失一般性，将狼群领地抽象为一个 $N\times m$ 的欧式空间，人工狼 i 的位置 $\boldsymbol{X}_i$ 表示为二进制编码 $(x_{i1},x_{i2},\cdots,x_{ij},\cdots,x_{im})$ $(i=1,2,\cdots,N;j=1,2,\cdots,m)$；$N$ 为人工狼总数；m 为编码长度。元素 x_{ij} 即为位置 $\boldsymbol{X}_i$ 的第 j 个编码位的值，且只能取 0 或 1。人工狼所感知的猎物气味浓度，即目标函数值为 $Y=f(\boldsymbol{X})$。并定义两人工狼 p 和 q 间的距离为两者二进制编码的 Manhattan 距离，如下式所示：

$$L(p,q)=\sum_{j=1}^{m}|x_{pj}-x_{qj}|,\ p,q\in\{1,2,\cdots,N\} \tag{3}$$

定义 1　反置。对 x_{ij} 反置即是对人工狼 i 的位置 $X_i=\{x_{i1},x_{i2},\cdots,x_{ij},\cdots,x_{im}\}$ 中第 j 个编码位的值 x_{ij} 进行赋值

$$x_{ij}=\begin{cases}1, & x_{ij}=0\\ 0, & x_{ij}=1\end{cases} \tag{4}$$

定义 2　运动算子。设人工狼 i 的位置为 $\boldsymbol{X}_i=\{x_{i1},x_{i2},\cdots,x_{im}\}$；$M$ 表示进行反置的编码位集合且不为空集，可理解为人工狼的可活动范围；r 表示进行反置的编码位的数目，可理解为人工狼的行走步长；运动算子 $\Theta(\boldsymbol{X}_i,M,r)$ 表示在 M 中随机选择 r 个编码位并将其值反置。例如

$$\boldsymbol{X}_i=\{1,0,0,1,0,0\},M=\{2,5\},r=1$$

则

$$\Theta(\boldsymbol{X}_i,M,r)=\{1,1,0,1,0,0\}\ \text{或}$$
$$\theta(\boldsymbol{X}_i,M,r)=\{1,0,\ 0,1,1,0\}$$

2.3　智能行为和规则的描述

BWP*A* 算法也由头狼产生规则，狼群更新机制和游走、召唤、围攻 3 种智能行为组成，由于前两者同文献[4]，在此不再赘述，下面就 3 种智能行为进行详细描述。

（1）游走行为。探狼 i 凭借敏锐的嗅觉，感知当前空气中的猎物气味浓度，即计算探狼 i 的目标函数值 Y_i。若 Y_i 大于 Y_{lead}，则令 $Y_{lead}=Y_i$，即探狼 i 成为头狼，其中 Y_{lead} 为头狼所感知到的气味浓度；若 $Y_i<Y_{lead}$，则探狼试探性地向 h 个方向前进一步侦查，即是将探狼位置 $\boldsymbol{X}_i$ 执行 h 次运动算子 $\Theta(\boldsymbol{X}_i,M,step_a)$，其中 $step_a$ 为游走步长，$M=\{1,2,\cdots,m\}$，同时，记录每次所感知的猎物气味浓度，而后探狼通过比较作出决策。若向第 p 个方向前进后所感知的猎物气味浓度为 Y_{ip}，$p\in H$，$H=\{1,2,\cdots,h\}$，则选择方向 p^* 前进一步并更新 $\boldsymbol{X}_i$，重复以上过程直至 $Y_i>Y_{lead}$ 或游走次数 T 超过限值 $T_{\max}$。其中选择的方向 p^* 满足

$$\begin{cases}p^*\in\max\{Y_{ip},\ p\in H\}\\ Y_{ip^*}>Y_{i0}\end{cases} \tag{5}$$

由于狼群中的探狼存在个体差异，因此其搜索猎物的方式也不尽相同，则 h 的取值也不同，具体可取 $[h_{\min},h_{\max}]$ 间的随机整数。

（2）召唤行为。头狼通过嚎叫发起召唤行为，指挥猛狼迅

速向其所在位置 $\boldsymbol{X}_d$ 靠拢；猛狼则自发地以某一相对较大的奔袭步长 $step_b$ 快速奔袭。即猛狼 i 的位置 $\boldsymbol{X}_i$ 依下式作变换：

$$X_i' = \Theta(\boldsymbol{X}_i, M, step_b) \tag{6}$$

式(6)中的 M 由下式求得：

$$M(k) = \begin{cases} j, k = k+1,\ j = j+1,\ x_{dj} \neq x_{ij} \\ null, k = k,\ j = j+1,\ x_{dj} = x_{ij} \end{cases} \tag{7}$$

式中，$j=1,2,\cdots,m$；k 的初值为 1；$null$ 表示空值。M 是猛狼位置 $\boldsymbol{X}_i$ 与头狼位置 $\boldsymbol{X}_d$ 不相同编码位的集合。对相同编码位的值不进行反置，体现猛狼的围猎基础和对自身优势的保持；对 $step_b$ 个相异编码位的值进行反置，表示人工狼逐渐向头狼位置聚集的趋势，体现头狼对狼群的指挥。但若 M 为空集，执行一次随机运动算子 $\Theta(\boldsymbol{X}_i, M^*, 1)$，此时，$M^* = \{1,2,\cdots,m\}$。

设 Y_i 为猛狼 i 所感知到的猎物气味浓度，若 $Y_i > Y_{lead}$，则 $Y_{lead} = Y_i$，猛狼 i 替代头狼；若 $Y_i < Y_{lead}$，则猛狼 i 继续奔袭直到 $d_{is} < d_{near}$ 时转入对猎物进行围攻。其中，d_{is} 为猛狼 i 与头狼间的距离；$d_{near} = \lceil m/w \rceil$ 为判定距离，w 为距离判定因子，$\lceil * \rceil$ 为向上取整。

(3) 围攻行为。为尽快将猎物捕杀，猛狼和探狼在头狼的指挥下对猎物施行围攻。具体地，将头狼所在位置 $\boldsymbol{X}_d$ 视为猎物的位置，参与围攻的人工狼 i 的位置 $\boldsymbol{X}_i$ 依下式进行位置变换：

$$\boldsymbol{X}_i' = \Theta(\boldsymbol{X}_i^*, M^*, step_c) \tag{8}$$

式中，$M^* = \{1,2,\cdots,m\}$；$\boldsymbol{X}_i^* = \Theta(\boldsymbol{X}_i, M, step_c)$，$M$ 的确定方式同式(7)；$step_c$ 为人工狼 i 执行围攻行为时的攻击步长。$\boldsymbol{X}_i^* = \Theta(\boldsymbol{X}_i, M, step_c)$ 体现了狼群的信息传递与共享机制，体现狼群中其他个体对群体优秀者，即对头狼的“追随”与“响应”；$\Theta(\boldsymbol{X}_i^*, M^*, step_c)$ 可理解为狼群在对猎物围攻过程中在猎物周围小范围内的群体运动，旨在防止狼群的“社会认知”和“个体认知”呈现高度趋同性，体现在优秀解域中对最优解的精细搜索，同时避免算法早熟。

比较人工狼实施围攻行为前后所感知到的猎物气味浓度并进行贪婪决策。

3 种智能行为中所涉及到游走步长 $step_a$、奔袭步长 $step_b$、攻击步长 $step_c$ 皆为整数，表示人工狼在解空间中搜寻最优解的精细程度。依据经验，长度为 m 的编码空间中存在如下关系：

$$\lceil m/10 \rceil \geqslant step_a \geqslant step_b \geqslant step_c = 1 \tag{9}$$

2.4 修复机制

在 BWPA 算法的狼群更新过程中会随机生成 R 匹新的人工狼，若新产生的人工狼不满足式(1)，则需要对其进行修复。可理解为狼群中的幼狼或病狼需要训练成长或康复后才能参加捕猎。对不满足式(1)的人工狼 $\boldsymbol{X}_i = \{x_{i1}, x_{i2}, \cdots, x_{ij}, \cdots, x_{im}\}$，重复执行运动算子 $\boldsymbol{X}_i^* = \Theta(\boldsymbol{X}_i, M, 1)$，直至满足。不满足背包容积限制意味着装入背包的物品过多，因此，M 由下式确定：

$$M(k) = \begin{cases} j, k = k+1,\ j = j+1,\ x_{ij} = 1 \\ null, k = k,\ j = j+1,\ x_{ij} = 0 \end{cases} \tag{10}$$

式中，$j=1,2,\cdots,m$；k 的初值为 1；$null$ 表示空值；k 的终值表示集合 M 包含的元素数目。

2.5 算法流程

由于在 BWPA 算法的实际编程计算时发现，当狼群中人工狼总量 N 一定时，人工探狼和人工猛狼数量比例较难确定，且多次计算发现两者都应尽量大，计算效果才佳。因此在具体计算中将人工探狼和猛狼的数量都设为 $N-1$，可理解为人工狼在不同的捕猎阶段承担不同的角色。综上，BWPA 算法的具体步骤如下。

步骤 1 初始化。设定狼群规模 N，各人工狼初始位置 $\{X_i\}$，算法的最大迭代次数为 $k_{\max}$，最大游走次数为 $T_{\max}$，步长为 $step_a$、$step_b$ 和 $step_c$，距离判定因子为 w，更新比例因子为 β。

步骤 2 最优人工狼被确定为头狼，其他人工狼为探狼并执行游走行为，直至 $Y_i > Y_{lead}$ 或 $T > T_{\max}$。

步骤 3 将除头狼外的狼视为猛狼并根据式(6)向猎物运动，若途中 $Y_i > Y_{lead}$ 则 $Y_{lead} = Y_i$，猛狼 i 替代头狼并发起召唤行为；若 $Y_i < Y_{lead}$，则猛狼持续奔袭直到 $d_{is} \leqslant d_{near}$。

步骤 4 根据式(8)更新参与围攻行为的人工狼的位置编码。

步骤 5 按头狼产生规则和狼群更新机制分别对头狼和狼群进行更新，并对不满足背包容积限制的人工狼位置编码进行修复。

步骤 6 判断是否达到终止条件，若达到则输出所求问题的最优解——头狼的位置编码 $\boldsymbol{X}_d$ 和头狼所感知的猎物气味浓度 Y_{lead}，否则转步骤 2。

3 算法理论分析

3.1 开拓能力和开采能力分析

学者们通常将算法在解空间中不断探索新解域的能力称为算法的开拓能力，将算法在某个小的解域内或其附近进行细致搜索的能力称为开采能力，任何一个较好的优化算法都应具备并处理好两者之间的平衡。BWPA 同时具备这两种能力。BWPA 通过步长来控制人工狼运动的速率，通过反置集合 M 来控制人工狼的活动范围，且行为设计和部分参数的选取都具有限定范围内随机变化的特点。这种随机性有别于其他智能算法如 BPSO 算法中粒子速度不断衰减的情况，能使人工狼的位置在整个求解过程中更为随机，进而在概率意义上遍历更多的解域，最终保证狼群具有较强的开拓能力。

具体地，游走行为对应较大的游走步长 $step_a$ 和最大的反置集合 M，使得 BWPA 算法在进化过程中能充分遍历搜索解空间，使算法具备较强的开拓能力；“胜者为王”的头狼产生规则和头狼的召唤行为使得狼群向最有可能捕获猎物的优良解域移动，使得算法的开拓更有针对性，大大增加了开采到最优解的概率；而围攻行为一般对应较小的攻击步长 $step_c$，并执行嵌入式运动算子 $\Theta(\boldsymbol{X}_i^*, M^*, step_c)$，$\boldsymbol{X}_i^* = \Theta(\boldsymbol{X}_i, M, step_c)$ 使得 BWPA 算法在优良解域深度开采最优解的同时具备在优良解域附近进行邻域开拓搜寻新的优质解域的能力，减小了算法陷入局部最优的概率；“强者生存”的狼群更新机制不断引入新的人工狼，这在很大程度上保证了狼群的多样性，使算法始终维持着较强的开拓能力，避免算法陷入局部最优；修复机制对不满足背包容积限制的人工狼的位置进行修复，减少了无用解对算法开拓和开采进程的影响。

综上所述，BWPA 继承了 WPA 算法基于职责分工的协作式寻优模式，通过设置合适的运动算子，能对 BWPA 算法的开拓和开采能力进行均衡，从而为实现较好的寻优效果提供条件。

3.2 收敛性分析

首先，人工狼的位置编码长度是有限的；其次，编码的每个位置的值仅能取 0 或 1，因此，整个解空间是有限的。而对于 0-1 背包这个具体问题，由于解空间中存在大量的不可行解，自然算法对应的可行解空间也是有限的。假设算法的最大迭代次数为无穷，BWPA 算法由于游走行为、围攻行为的随机性及每代的“强者生存”的狼群更新机制都表明狼群中的人工狼位置的编码位一直具备发生变动的能力和不断在优秀的解域开拓新解的能力。因此，对于有限可行解空间，算法经历无穷次迭代后，狼群中曾出现的人工狼个体的位置必定能遍历到所有可行解；同时由于 BWPA 算法中头狼不经历游走、召唤和围攻等智能行为而直接进入下一代，直到它被更强的人工狼所取代，始终保存了当前的最优值到下一代种群，直至算法结束。因此可知，BWPA 算法在理论上具有全局收敛性。

但实际情况并不然，同其他仿生进化算法一样，由于不可能使得算法进行无穷次迭代计算，因此各算法在实际的寻优计算中仍有可能会陷入局部最优。根据文献[10]的研究可知：仿生算法最终能否获得全局极值的关键在于在整个求解过程中，种群有没有从解集合中挑选出最优群体，以及挑出最优个体的概率。BWPA 算法基于狼群职责分工的协作式搜索模式，且经合理设置不同阶段的运动算子 $\Theta(X_i, M, r)$ 和狼群的更新规则可实现对 0-1 背包问题等现实问题的有效抽象和建模，有利于对全局极值的快速搜寻，减小算法陷入局部极值的概率。这点将在下面的实验与分析部分进行详细说明。

4 实验与分析

4.1 10 个经典的背包问题算例(实验 1)

如表 1 所示，为充分测试 BWPA 算法的性能和特点，引入被广泛使用的 10 个经典 0-1 背包问题来测试算法的寻优性能。其中 $k1 \sim k4$，$k5 \sim k7$，$k8$，$k9 \sim k10$ 分别来自文献[12]～文献[15]，且其维数 m 从 10 维至 100 维，可较全面地反应算法的性能。表 1 中已知最优解均采用 A/B 的形式，A 为背包中物品总价值，B 为背包中物品总体积。

表 1　0-1 背包问题的 10 个仿真实例

编号	维数	参数(w, p, C)	已知最优解
$k1$	10	w=(95,4,60,32,23,72,80,62,65,46)，p=(55,10,47,5,4,50,8,61,85,87)，C=269	295/269
$k2$	15	w=(56.358 531,80.874 050,47.987 304,89.596 240,74.660 48,85.894 345,51.353 496,1.498 459,36.445 204,16.589 862,44.569 23,0.466 9,37.788 018,57.118 442,60.716 575)，p=(0.125 126,19.330 424,58.500 931,35.029 145,82.284 005,17.410 810,71.050 142,30.399 487,9.140 294,14.731 285,98.852 504,11.908 322,0.891 140,53.166 295,60.176 397)，C=375	481.07/354.96
$k3$	20	w=(92,4,43,83,84,68,92,82,6,44,32,18,56,83,25,96,70,48,14,58)，p=(44,46,90,72,91,40,75,35,8,54,78,40,77,15,61,17,75,29,75,63)，C=878	1 024/871
$k4$	23	w=(983,982,981,980,979,978,488,976,972,486,486,972,972,485,485,969,966,483,964,963,961,958,959)，p=(981,980,979,978,977,976,487,974,970,485,485,970,970,484,484,976,974,482,962,961,959,958,857)，C=10 000	9 767/9 768
$k5$	50	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)，p=(220,208,198,192,180,180,$\overline{165}$,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)，C=1 000	3 103/1 000
$k6$	50	p 和 C 同上，w 除第 29 件物品的体积由“50”变为“25”外其他皆同上	3 119/1 000
$k7$	50	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)，p=(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)，C=11 258	16 102/11 231
$k8$	60	w=(135,133,130,11,128,123,20,75,9,66,105,43,18,5,37,90,22,85,9,80,70,17,60,35,57,35,61,40,8,50,32,40,72,35,100,2,7,19,28,10,22,27,30,88,91,47,68,108,10,12,43,11,20,37,17,4,3,21,10,67)，p=(350,310,300,295,290,287,283,280,272,270,265,251,230,220,215,212,207,203,202,200,198,196,190,182,181,175,160,155,154,140,132,125,110,105,101,92,83,77,75,73,72,70,69,66,60,58,45,40,38,36,33,31,27,23,20,19,10,9,4,1)，C=2 400	8 362/2 393
$k9$	80	w=(40,27,5,21,51,16,42,18,52,28,57,34,44,43,52,55,53,42,47,56,57,44,16,2,12,9,40,23,56,3,39,16,54,36,52,5,53,48,23,47,41,49,22,42,10,16,53,58,40,1,43,56,40,32,44,35,37,45,52,56,40,2,23,49,50,26,11,35,32,34,58,6,52,26,31,23,4,52,53,19)，p=(199,194,193,191,189,178,174,169,164,164,161,158,157,154,152,152,149,142,131,125,124,124,124,122,119,116,114,113,111,110,109,100,97,94,91,82,82,81,80,80,80,79,77,76,74,72,71,70,69,68,65,65,61,56,55,54,53,47,47,46,41,36,34,32,32,30,29,29,26,25,23,22,20,11,10,9,5,4,3,1)，C=1 173	5 183/1 170
$k10$	100	w=(54,95,36,18,4,71,83,16,27,84,88,45,94,64,14,80,4,23,75,36,90,20,77,32,58,6,14,86,84,59,71,21,30,22,96,49,81,48,37,28,6,84,19,55,88,38,51,52,79,55,70,53,64,99,61,86,1,64,32,60,42,45,34,22,49,37,33,1,78,43,85,24,96,32,99,57,23,8,10,74,59,89,95,40,46,65,6,89,84,83,6,19,45,59,26,13,8,26,5,9)，p=(297,295,293,292,291,289,284,284,283,283,281,280,279,277,276,275,273,264,260,257,250,236,236,235,235,233,232,232,228,218,217,214,211,208,205,204,203,201,196,194,193,193,192,191,190,187,187,184,184,184,181,179,176,173,172,171,160,128,123,114,113,107,105,101,100,100,99,98,97,94,94,93,91,80,74,73,72,63,63,62,61,60,56,53,52,50,48,46,40,40,35,28,22,22,18,15,12,11,6,5)，C=3 820	15 170/3 818

计算时，为充分测试 BWPA 算法的性能，限制狼群规模和最大迭代次数皆为 $4m$。对上述背包问题，分别运算 20 次。如表 2 所示，从最优值、最差值、平均值等多方面对算法进行评估。其中命中次数指 20 次计算中得到已知最优值的次数。表 2 同时给出其他参考文献所提供的最优解，采用 A/B_C[D] 的形式，A、B 的含义同上，C 为所用方法，[D]为来源的文献，“－”表示文献未提供。

表 2 0-1 背包问题的 10 个仿真实例的结果对比

编号	维数	BWPA 的解					其他文献提供的最优解
		最优解	最差值	平均值	标准差	命中次数	
$k1$	10	295/269	295/269	295	0	20	294/260_遗传算法[16]，290/237_贪婪算法[16]，295/269_模糊粒子群算法[16]
$k2$	15	481.07/354.96	481.07/354.96	481.07	0	20	481.07/354.96_自适应和声搜索算法[8]
$k3$	20	1 024/871	1 024/871	1 024	0	20	1 013/851_粒子群算法[16]，1 018/837_贪婪算法[16]，1 024/871_量子和声搜索算法[17]
$k4$	23	9 767/9 768	9 767/9 768	9 767	0	20	9 757/9 777_降维替换算法[18]，9 767/9 768_量子和声搜索算法[17]
$k5$	50	3 103/1 000	3 097/1 000	3 102.25	1.86	17	3 075/－_基本人工鱼群算法[19]，3 082/－_模拟退火算法[20]，3 090/－_基于模拟退火的遗传算法[19]，3 098/－_标准蚁群算法[13]，3 103/1 000_病毒协同进化粒子群算法[13]
$k6$	50	3 119/1 000	3 114/1 000	3 118.35	1.53	15	3 105/997_基于混合编码差异演化算法[21]，3 112/1 000_贪心遗传算法[22]，3 114/1 000_学习型和声搜索算法[23]，3 119/1 000_基因属性保留遗传算法[24]
$k7$	50	16 102/11 231	16 102/11 231	16 102	0	20	14 865/－_遗传算法[13]，15 565/－_二进制粒子群算法[25]，15 955/－_模拟退火算法[13]，16 052/－_二进制混合粒子群算法[25]，16 102/11 231_病毒协同进化粒子群算法[13]
$k8$	60	8 362/2 393	8 356/2 395	8 361.40	1.85	17	7 775/2 371_基于贪心策略改进的遗传算法[14]，8 362/－_引入侦查子群的蚁群算法[14]
$k9$	80	5 183/1 170	5 183/1 170	5 183	0	20	5 107/1 167_混合粒子群算法[15]，5 107/1 172_基于罚函数策略的离散粒子群算法[15]，5 183/1 170_二进制杂草算法[10]
$k10$	100	15 170/3 818	15 170/3 818	15 170	0	20	15 080/3 819_混合离散粒子群算法[15]，15 089/3 817_基于罚函数策略的离散粒子群算法[15]，15 170/3 818_基于贪心变换策略的离散粒子群算法[15]

由表 2 可以看出，BWPA 对于 10 个 0-1 背包问题都具有较好的寻优效果，都能得到已知最优解且计算稳定性较好。BWPA 所得结果比很多文献所提算法都要更优或与新近提出的如二进制杂草算法、量子和声搜索等算法的寻优效果相当。特别需要说明的是 $k5$ 和 $k6$ 两个算例，两者的差别仅在第 29 件物品的体积，前者为 50，后者为 25，两者的最优解也不同，分别为 3 103/1 000 和 3 119/1 000。$k6$ 源于 $k5$，而 $k5$ 源于文献[26]。很多文献将 $k6$ 的结果同 $k5$ 的结果相比较[27-28]，这显然有失偏颇。从表 2 也可以看出，虽然在多次计算中 BWPA 偶尔也会陷入局部极值，但即使其得到的最差值也依然优于很多文献所提算法得到的最优解。同时对于 $k1$、$k2$、$k3$、$k4$、$k7$、$k9$、$k10$ 这 6 个背包问题，特别是 $k7$、$k9$、$k10$ 3 个背包问题的维数已分别达到 50 维、80 维、100 维，BWPA 在 20 次计算中每次都能得到已知最优解，充分显示 BWPA 算法在离散空间中有较强的全局寻优能力和计算鲁棒性。

4.2 高维背包问题仿真实验(实验 2)

这里采用文献[26]所定义的背包问题进行测试。第 $i(i=1,2,\cdots,m)$ 件物品的体积 w_i，价值 p_i 及背包容量 C 由以下公式随机产生：

$$w_i = \text{randint}[1,10]; p_i = w_i + 5; C = (1/2)\sum_{i=1}^{m} w_i \tag{11}$$

式中，randint[1,10]表示随机取{1,2,…,10}范围内的一个整数；m 为待装物品总数，即背包问题的维数和人工狼的位置编码长度。分别产生 100 维、250 维、500 维的 3 个高维背包问题算例，每个算例在随机产生后就保持不变。利用经典的离散粒子群算法(discrete particle swarm optimization algorithm，DPSO)[9]、量子遗传算法(quantum genetic algorithm，QGA)[30]、贪婪遗传算法(greedy genetic algorithm，GGA)[22]与 BWPA 算法进行对比验证。

为充分测试各算法性能，设置最大迭代次数、各算法初始种群规模皆为 $2m$，其他参数根据相应文献推荐的最优参数设置。具体地，BWPA 算法中设置最大游走次数 $T_{\max}=10$，距离判定因子 $w=m/4$，各步长 $step_a/4=step_b/8=step_c=1$，更新比例因子 $\beta=4$；DPSO 算法学习因子 $c_1=c_2=1.496\ 2$，惯性权重 $w=0.729\ 8$；GGA 算法交叉概率和变异概率分别为 $p_c=0.5$ 和 $p_m=0.01$，采用多点交叉和变异，运用轮盘赌法进行选择操作，并运用价值体积比的贪婪策略；QGA 算法二进制编码长度为 20。用上述 4 种方法分别对 3 个高维

KP 问题进行 20 次寻优计算。如图 1 所示，首先从各算法 20 次计算得到的最佳物品总价值的分离线直观地看，背包问题维数越高，BWPA 的优化效果越明显，在图 1 中表现为随着维数 m 的增加，各数据线的分离性逐渐增大。

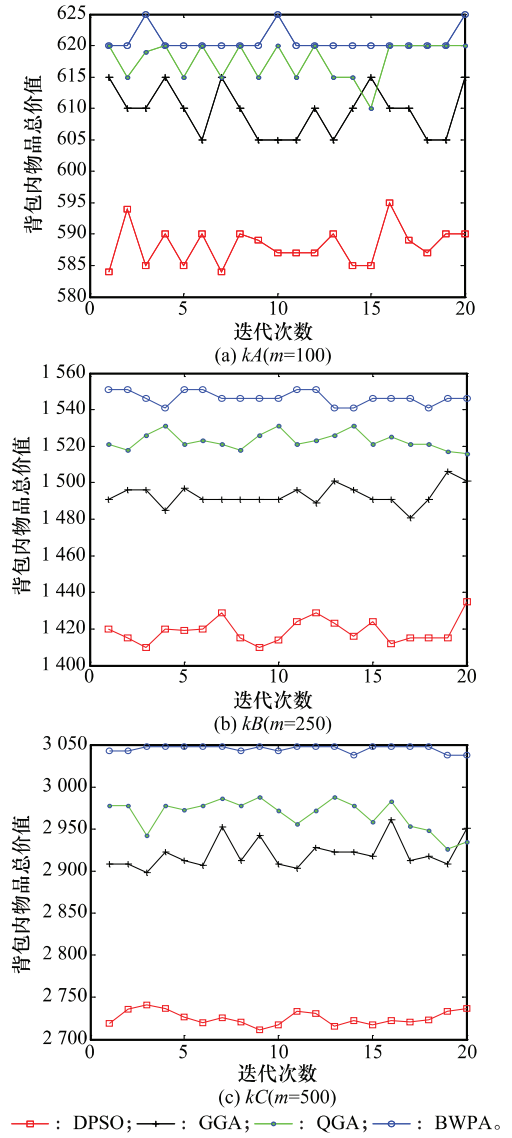

图 1　各算法对不同背包问题求解所得物品总价值的分离线图

表 3 为各算法的求解结果，分别从最优值、最差值、平均值和标准差 4 个方面进行全面评估。其中最优解、平均值可体现算法的收敛精度和寻优能力，最差解、标准差体现了算法的稳定性和对抗局部极值的能力。当 $m=100$ 时，除 DPSO 外，BWPA、GGA、QGA 这 3 种算法无论是求解质量还是求解的稳定性差别并不大；但随着背包维数 m 的增加，解空间呈几何倍增长，各算法的求解质量和稳定性都受到一定影响。特别当 $m=500$ 时，搜索空间达到 2^{500} 级，面对如此庞大的解空间，算法在有限迭代次数内仅能开采较少的解域，造成求解质量的下降，这就对算法的寻优模式提出了较高的要求。但 BWPA 算法的最差解 3 038/1 308 仍然优于其他 3 种算法的最优解，如 DPSO 算法的 2 741/1 273，QGA 算法的 2 988/1 308，GGA 算法的 2 961/1 306。同时，随着背包规模的增大，各算法求解的稳定性也会下降，相比之下 QGA 和 GGA 2 算法的标准差增加更为明显，当 $m=500$ 时，标准差分别上升至 18.35 和 17.62。而 BWPA 算法的标准差始终在 4 以内，体现其算法的鲁棒性、求解的稳定性和寻优模式的优势。

表 3　3 个高维 0-1 背包问题的结果对比

编号	维数	算法	最优解	最差解	平均值	标准差
kA	100	BWPA	625/285	620/285	620.75	1.83
		DPSO	595/303	584/280	588.15	3.10
		QGA	620/285	610/285	617.70	2.99
		GGA	615/285	605/285	609.50	3.94
kB	250	BWPA	1 551/691	1 541/691	1 546.50	3.59
		DPSO	1 435/639	1 414/596	1 419.00	6.68
		QGA	1 531/691	1 516/691	1 522.90	4.51
		GGA	1 506/691	1 481/691	1 493.15	5.66
kC	500	BWPA	3 048/1 308	3 038/1 308	3 045.50	3.80
		DPSO	2 741/1 273	2 711/1 141	2 725.90	7.94
		QGA	2 988/1 308	2 926/1 306	2 967.40	18.35
		GGA	2 961/1 306	2 898/1 308	2 920.90	17.62

图 2 为采用 4 种算法对 3 个高维背包问题 20 次独立求解计算的平均进化曲线。

分析认为，DPSO 算法具有运算简单、易于实现、鲁棒性较好等优点，因而被广泛应用。但 DPSO 算法中粒子位置的更新主要通过比较自身位置与其周围位置和群粒子中当前最优位置来进化的，模式较为单一，使得其收敛速度在计算初期较快，但在计算的后期效率不高，也易陷入局部极值[31]。因此，在表 3 中表现为 DPSO 取得的最优值、最差值、平均值较差，但标准差却小于 GGA 算法；在图 2 中表现为前期迭代收敛较快，但中后期粒子探索新的更优解能力却很弱，如图 2(c)中 DPSO 算法经过 100 次迭代后即陷入局部最优 2 707，而在后 900 次迭代中陷入此局部最优解未能跳出，导致最终所得的平均最优值较差。

遗传算法具有简单通用、适于并行分布处理等特点，GGA 算法采用了价值体积比的贪婪策略对遗传算法进行改进，改善了种群的个体质量，有利于背包问题的求解。但求解到一定范围时，GA 算法可能会出现大量无效的迭代计算，大大降低了算法的寻优效率，同时也使得算法易陷入局部极值[32]。QGA 算法优势在于量子并行性，使得每个量子比特编码组成的染色体均包含了更多的子空间，种群多样性更好。因此，QGA 算法的搜索范围要比 GGA 算法更广，具有更强的并行搜索能力，相对而言，QGA 的寻优效果要优于 GGA 算法[33]。在表 3 中表现为 QGA 算法能获得相对更优的最优解和平均值，但 QGA 算法的本质是在连续空间中寻优，致使其生成的连续解与问题的目标函数之间存在多对一的映射，这就导致大量冗余解和重复搜索，致使收敛效率不高。如图 2 所示，在限定最大迭代次数的情

况下，QGA 的平均进化曲线始终在 BWPA 算法的曲线之下，寻优效果劣于直接在离散空间中利用 BWPA 算法。并且，这种趋势随着背包规模的增大而越发凸显。

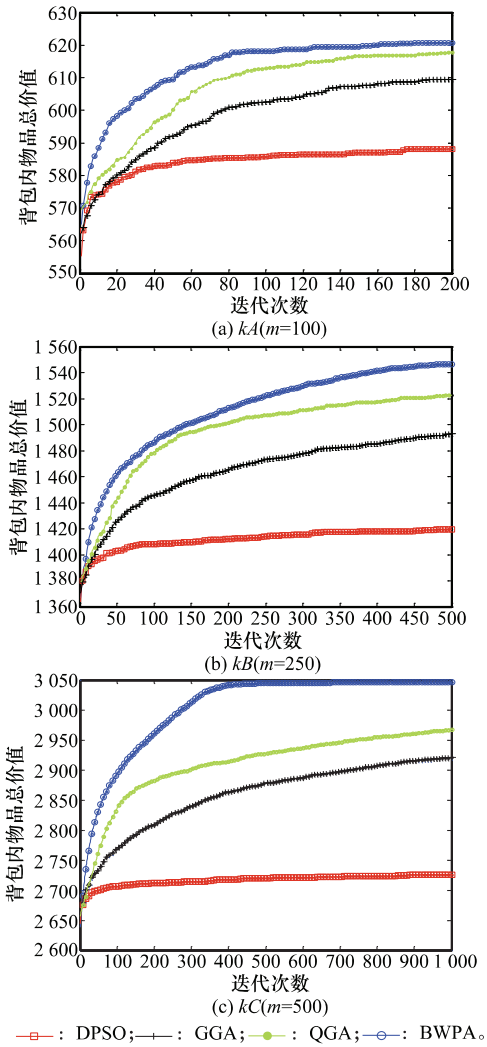

图 2 不同算法的平均进化曲线

再如表 3 所示，BWPA 算法对各问题所求的最优解、最差解和平均值全部优于其他 3 种算法，标准差也较低，且随着问题规模的增大 BWPA 的优势更为明显；图 2(a)中，搜索前期各算法都具有较快的收敛速度，但当平均进化曲线进入平缓阶段之后，BWPA 所得的曲线仍在其他 3 种算法的曲线之上；由图 2(b)可见，BWPA 算法在搜索后期依然能够搜寻到质量更好的非劣解，具有持续搜索的能力；图 2(b)和图 2(c)中，特别对于 kC 问题，BWPA 具有快速收敛能力，即经历过 m 次迭代后都可以得到全局最优解或较好的次优解。另外，直观对比图 2的 3 幅图也可看出，随着背包问题规模的增大，BWPA 所得的平均进化曲线与其他 3 条进化曲线的差异逐渐增大，可见 BWPA 算法在解决高维背包问题时的相对优势。

5 结 论

在狼群算法的基础上，引入二进制编码，提出了一种用于求解 0-1 背包问题的二进制狼群算法。通过 10 个经典的 0-1 背包算例和 3 个高维 0-1 背包问题的仿真对比研究，表明 BWPA 算法具有相对较好的求解稳定性、收敛性和全局搜索能力，尤其在求解大规模 0-1 背包问题时优势明显，优于经典的 DPSO 算法、GGA 算法和 QGA 算法。将来可进一步分析 BWPA 算法的部分参数效应，并将其应用于任务调度、资源分配、材料切割等组合优化问题中。

参考文献：

[1] Yang C G, Tu X Y, Chen J. Algorithm of marriage in honey bees optimization based on the wolf pack search[C]// *Proc. of the International Conference on Intelligent Pervasive Computing*, 2007:462-467.

[2] Liu C G, Yan X H, Liu C Y, et al. The wolf colony algorithm and its application[J]. *Chinese Journal of Electronics*, 2011, 20(2): 212-216.

[3] Rui T, Fong S, Yang X, et al. Wolf search algorithm with ephemeral memory[C]// *Proc. of the Seventh International Conference on Digital Information Management*, 2012:165-172.

[4] Wu H S, Zhang F M, Wu L S. New swarm intelligence algorithm-wolf pack algorithm[J]. *Systems Engineering and Electronics*, 2013, 35(11): 3430-3438.(吴虎胜，张凤鸣，吴庐山. 一种新的群体智能算法——狼群算法[J]. 系统工程与电子技术，2013,35(11):3430-3438.)

[5] Jagdish C B, Kusum D. A modified binary particle swarm optimization for knapsack problems[J]. *Applied Mathematics and Computation*, 2012,218(22):11042-11061.

[6] Zou D X, Gao L Q, Li S, et al. Solving 0-1 knapsack problem by a novel global harmony search algorithm[J]. *Applied Soft Computing*, 2011,11(2):1556-1564.

[7] Julien R, Jose R F, Yves D S. The inverse {0,1}-knapsack problem: theory, algorithms and computational experiments[J]. *Discrete Optimization*, 2013,10(2):181-192.

[8] Zhang X G, Huang S Y, Hu Y, et al. Solving 0-1 knapsack problems based on amoeboid organism algorithm[J]. *Applied Mathematics and Computation*, 2013,219(19):9959-9970.

[9] Kennedy J, Eberhart R C. A discrete binary version of the particle swarm algorithm[C]// *Proc. of the World Conference on Systemics, Cybernetics and Informatics*, 1997:4104-4109.

[10] Zhang S, Wang Y G, Xia L N. Invasive weed optimization algorithm of discrete binary version[J]. *Journal of Huazhong University of Science and Technology (Natural Science Edition)*, 2011,39(10):55-60.(张帅，王营冠，夏凌楠. 离散二进制入侵杂草算法[J]. 华中科技大学学报(自然科学版)，2011,39(10): 55-60.)

[11] Wang L, Yang R X, Xu Y, et al. An improved adaptive binary harmony search algorithm[J]. *Information Sciences*, 2013, 232 (12):58-87.

[12] Qian J, Zheng J G. Greedy quantum inspired evolutionary algorithm for quadratic knapsack problem[J]. *Computer Integrated Manufacturing Systems*, 2012, 18 (9): 2003-2011.(钱洁，郑建国. 二次背包问题的贪婪量子进化算法求解[J]. 计算机集

成制造系统，2012，18(9)：2003-2011.)

[13] Gao F，Cui G，Wu Z B，et al. Virus-evolutionary particle swarm optimization algorithm for knapsack problem[J]. *Journal of Harbin Institute of Technology*，2009，41(6)：103-107.(高芳，崔刚，吴智博，等. 求解背包问题的病毒协同进化粒子群算法[J]. 哈尔滨工业大学学报，2009，41(6)：103-107.)

[14] Hu Z H，Zhao M. Using ant colony optimization algorithm with scout subgroup to solve 0/1 knapsack problem[J]. *Journal of Guizhou Normal University*(*Natural Sciences*)，2009，27(3)：82-88.(胡中华，赵敏. 引入侦查子群的蚁群算法求解 0/1 背包问题[J]. 贵州师范大学学报(自然科学版)，2009，27(3)：82-88.)

[15] Liu J Q，He Y C，Gu Q Q. Solving knapsack problem based on discrete particle swarm optimization[J]. *Computer Engineering and Design*，2007，28(13)：3189-3191.(刘建芹，贺毅朝，顾茜茜. 基于离散微粒群算法求解背包问题研究[J]. 计算机工程与设计，2007，28 (13)：3189-3191.)

[16] Liu Y，Ma L. Solving 0-1 knapsack problem by fuzzy particle swarm optimization[J]. *Application Research of Computers*，2011，28 (11)：4026-4031.(柳寅，马良. 0-1 背包问题的模糊粒子群算法求解[J]. 计算机应用研究，2011，28(11)：4026-4031.)

[17] Layeb A. A hybrid quantum inspired harmony search algorithm for 0-1 optimization problems[J]. *Journal of Computational and Applied Mathematics*，2013，253(12)：14-25.

[18] Gao T，Wang M G，Tang L X，et al. The research for the reductive dimension and replacive variable algorithm of special restrict 0-1 ILP[J]. *Systems Engineering-Theory Methodology Applications*，2002，11(2)：125-130.(高天，王梦光，唐立新，等. 特殊一维背包问题的降维替换算法研究[J]. 系统工程理论方法与应用，2002，11(2)：125-130.)

[19] She X Y，Zhu M H，Zhao Y M. Improved artificial fish school algorithm to solve knapsack problem[J]. *Computer Engineering and Applications*，2011，47(21)：43-46.(库向阳，朱命昊，赵亚敏. 求解 0/1 背包问题的改进人工鱼群算法研究[J]. 计算机工程与应用，2011，47(21)：43-46.)

[20] Zhang S Y，Cai Z H，Zhan Z G. Solving 0-1 knapsack problem based on genetic algorithm with improved simulated annealing [J]. *Microelectronics & Computer*，2011，28(2)：61-64.(张盛意，蔡之华，占志刚. 基于改进模拟退火的遗传算法求解0-1背包问题[J]. 微电子学与计算机，2011，28(2)：61-64.)

[21] Deng C S，Zhao B Y，Liang C Y. Mixed-coding-based differential evolution algorithm for 0-1 knapsack problem[J]. *Application Research of Computers*，2010，27(6)：2031-2033.(邓长寿，赵秉岩，梁昌勇. 基于混合编码的差异演化算法解 0-1 背包问题[J]. 计算机应用研究，2010，27(6)：2031-2033.)

[22] He Y C，Liu K Q，Zhang C J，et al. Greedy genetic algorithm for solving knapsack problems and its applications[J]. *Computer Engineering and Design*，2007，28 (11)：2655-2657.(贺毅朝，刘坤起，张翠军，等. 求解背包问题的贪心遗传算法及其应用[J]. 计算机工程与设计，2007，28(11)：2655-2657.)

[23] Li R P，Ouyang H B，Gao L Q，et al. Learned harmony search algorithm and its application to 0-1 knapsack problems[J]. *Control and Decision*，2013，28(2)：205-210.(李若平，欧阳海滨，高立群，等. 学习型和声搜索算法及其在 0-1 背包问题中的应用[J]. 控制与决策，2013，28(2)：205-210.)

[24] Ma F N，Xie L，Zheng Z. Attribute gene-reserved genetic algorithm for solving knapsack problem[J]. *Journal of Tianjin University*，2010，43(11)：1020-1024.(马丰宁，谢龙，郑重. 求解背包问题的基因属性保留遗传算法[J]. 天津大学学报，2010，43(11)：1020-1024.)

[25] Luo J W. Binary hybrid particle swarm optimization algorithm base on crossover operation for solving knapsack problem[J]. *Journal of Central South University of Forestry & Technology*，2011，31 (9)：170-174.(罗健文. 基于交叉操作的二进制混合粒子群算法求解背包问题[J]. 中南林业科技大学学报，2011，31(9)：170-174.)

[26] Li J，Fang P，Zhou M. A hybrid genetic algorithm for knapsack problem[J]. *Journal of Nanchang Institute of Aeronautical Technology*，1998(3)：31-35.(李娟，方平，周明. 一种求解背包问题的混合遗传算法[J]. 南昌航空工业学院学报，1998(3)：31-35.)

[27] He Y C，Wang X Z，Kou Y Z. A binary differential evolution algorithm with hybrid encoding[J]. *Journal of Computer Research and Development*，2007，44(9)：1476-1484.(贺毅朝，王熙照，寇应展. 一种具有混合编码的二进制差分演化算法[J]. 计算机研究与发展，2007，44(9)：1476-1484.)

[28] Tuo S H，Zhou T. New algorithm of solving 0-1 knapsack problem based on dynamic telescopic strategy[J]. *Computer Engineering and Applications*，2012，48(4)：47-49.(拓守恒，周涛. 一种用于求解 0-1 背包问题的动态伸缩算法[J]. 计算机工程与应用，2012，48(4)：47-49.)

[29] Truong T K，Li K L，Xu Y M. Chemical reaction optimization with greedy strategy for the 0-1 knapsack problem[J]. *Applied Soft Computing*，2013，13(4)：1774-1780.

[30] Han K H. Genetic quantum algorithm and its application to combinatorial optimization problem[C]// *Proc. of the Congress on Evolutionary Computation*，2000：1354-1360.

[31] Trelea I C. The particle swarm optimization algorithm：convergence analysis and parameter selection[J]. *Infomation Processing Letters*，2003，85(6)：317-325.

[32] Yü S W，Wei Y M，Zhu K J. Hybrid optimization algorithms based on particle swarm optimization and genetic algorithm[J]. *Systems Engineering and Electronics*，2011，33 (7)：1647-1652.(於世为，魏一鸣，诸克军. 基于粒子群-遗传的混合优化算法[J]. 系统工程与电子技术，2011，33(7)：1647-1652.)

[33] Zakaria L，Salim C. Comparison of genetic algorithm and quantum genetic algorithm[J]. *The International Arab Journal of Information Technology*，2012，9(3)：243-249.

作者简介：

吴虎胜(1986-)，男，博士研究生，主要研究方向为信息系统工程与智能决策、智能数据挖掘。

E-mail：wuhusheng0421@163.com

张凤鸣(1963-)，男，教授，博士，主要研究方向为系统工程、数据挖掘、故障诊断。

E-mail：zfmwenzhang007@163.com

战仁军(1963-)，男，教授，博士，主要研究方向为军事装备学、警用器材、非致命武器。

E-mail：zrwenzhang007@163.com

张　超(1986-)，男，博士研究生，主要研究方向为信息系统工程与智能决策。

E-mail：25728042@163.com