

基于改进蝙蝠算法的混合整数规划问题

赵乃刚, 李 勇

(山西大同大学 数学与计算机科学学院, 山西 大同 037009)

摘 要: 针对非线性混合整数规划问题, 提出了一种改进的蝙蝠算法. 构造出一种自适应调整的局部搜索步长, 同时对整数变量采用单位步长搜索, 以此逐步提高蝙蝠算法的局部开发能力; 引入自然选择原理, 平衡改进蝙蝠算法的全局搜索能力; 初始一个可行解, 保证算法的正确搜索方向. 通过 13 个常见的测试函数测试结果表明, 改进的蝙蝠算法对求解非线性混合整数规划问题, 在成功率和精度方面都不亚于改进的粒子群算法.

关键词: 蝙蝠算法; 非线性混合整数规划; 自适应搜索步长; 自然选择

中图分类号: TP18

文献标识码: A

文章编号: 1000-7180(2017)06-0094-05

An Improved Bat Algorithm for Solving Nonlinear Mixed Integer Optimization Problems

ZHAO Nai-gang, LI Yong

(School of Mathematics and Computer Science, Shanxi Datong University, Datong 037009, China)

Abstract: This paper proposed an improved BA algorithm for solving nonlinear mixed integer optimization problems (NMIP) problem. An adaptive local search step size is constructed, and the integer variable is searched by unit step size, this improves the local search ability of the improved algorithm. Natural selection principle can improve the global search ability of the improved algorithm. The improved algorithm uses a feasible solution to ensure the correct search direction of the algorithm. Experimental results of sixteen test function show that the improved bat algorithm is not inferior to the improved particle swarm optimization algorithm for solving nonlinear mixed integer optimization problems.

Key words: bat algorithm(BA); nonlinear mixed integer programming(NMIP); adaptive search step; natural selection

1 引言

非线性混合整数规划(NMIP)问题是指目标函数和约束函数为非线性的, 变量是连续实数和离散整数的数学规划问题. NMIP 问题是一类 NP 难问题, 本文研究的非线性混合整数规划(NMIP)问题的一般模型为

$$\begin{aligned} \min \quad & F = f(x, y) \\ \text{s. t.} \quad & h_i(x, y) = 0 \quad i = 1, 2, \dots, m \\ & g_j(x, y) \leq 0 \quad j = 1, 2, \dots, l \\ & x \in X, y \in Y \end{aligned} \tag{1}$$

式中, $h_i(x, y)$ 为等式约束; $g_j(x, y)$ 为不等式约束; x 为实数型变量; y 为整数型变量; m 和 l 分别为等式约束和不等式约束的个数.

求解 NMIP 问题至今有两类方法: 一类是运用传统算法进行求解, 一类是运用智能优化算法进行求解. 传统算法包括分支定界法、割平面法、外逼近法等^[1-2]. 传统算法的确定性使求解 NMIP 问题上存在局限性和不足. 另一类算法主要包括遗传算法、蚁群算法、粒子群算法、差分进化算法等智能算法^[3-5].

蝙蝠算法(BA)是在 2010 年由 Yang^[6] 模拟蝙

蝙蝠觅食行为提出来的一种新型算法. 该算法设置参数少、实现方式简单,自提出以来,已逐渐被诸多学者成功应用于一些优化问题中^[7-8],但还没有运用在非线性混合整数规划问题上.

本文提出了一种改进的蝙蝠算法来求解非线性混合整数规划问题. 在本算法中,给出了一种自适应变化扰动步长,使算法能够更好地找出局部精确解,同时采用自然选择策略更新最差的部分解,提高种群的多样性.

2 基本蝙蝠算法

蝙蝠一般在夜间活动,在黑暗中发出声音脉冲,根据接收到的回声来探索食物的位置、避开障碍物及寻找栖息地. 随着与搜索猎物越来越近,蝙蝠发出声音的响度从也越来越小,直至到靠近猎物时的无声. 蝙蝠根据声音的发出和接收时间,来判断与目标的距离、方位及移动速度. 将蝙蝠的寻找食物过程设计如下:

(1) 蝙蝠都能够利用自己生理特性感知目标距离,而且能区分食物与障碍物;

(2) 蝙蝠位置 X_i 以速度 V_i 进行飞行,通过频率 f_i 、化波长 λ 和响度 A_0 来搜索目标;并且通过目标和自身的距离变化来调节下一次发射出的脉冲波长(或频率);

(3) 响度的变化是从最大值 A_0 到最小值 $A_{\min}$.

由此概括蝙蝠算法的更新规则:

假设蝙蝠在 d 维空间中飞行搜索, $t-1$ 时刻,蝙蝠 i 的位置为 X_i^{t-1} , 飞行速度为 V_i^{t-1} , 全局最优位置为 X_* , 则蝙蝠 i 在 t 时刻的位置和飞行速度如下:

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta \quad (2)$$

$$V_i^t = V_i^{t-1} + (X_i^{t-1} - X_*)f_i \quad (3)$$

$$X_i^t = X_i^{t-1} + V_i^t \quad (4)$$

对于局部搜索,在 $[0, 1]$ 之间产生一个随机数 $rand1$, 如果 $rand1 > r_i^t$, 则对当前最优解按式(5)进行随机扰动,产生一个新解 X_{new} , 并对新的解进行越界处理. 再产生一个随机数 $rand2$, 如果 $rand2 > A_i^t$ 且 $f(X_{\text{new}}) < f(X_*)$, 则蝙蝠 i 移向 X_{new} , 同时按式(6)和(7)进行调整脉冲响度 A_i^t 和速率 r_i , 使蝙蝠从发现目标开始逐渐降低其脉冲发射响度 A_i^t , 且逐渐提高其脉冲发射的速率 r_i .

$$X_{\text{new}} = X_* + \epsilon A^t \quad (5)$$

$$A_i^t = \alpha A_i^{t-1} \quad (6)$$

$$r_i^t = r_i^0 [1 - e^{-\gamma}] \quad (7)$$

式中, $f_{\min}$ 和 $f_{\max}$ 分别是蝙蝠发出声波的最小和最

大频率; β 为 $[0, 1]$ 之间的均匀随机数; 每只蝙蝠发射声波的频率初始值在范围 $[f_{\min}, f_{\max}]$ 内均匀随机分布; ϵ 为 $[-1, 1]$ 中的随机数; $A^t = \langle A_i^t \rangle$ 为全体蝙蝠在迭代次数为 t 时的平均响度; r_i^0 为初始速率; α 和 γ 均为常数, $0 < \alpha < 1, 0 < \gamma < 1$.

3 混合整数规划问题的改进蝙蝠算法

3.1 种群初始化

对于实数变量 x_i 和整数变量 y_i 分别根据下面公式生成:

$$x_i = x^L + rand * (x^U - x^L) \quad (8)$$

$$y_i = round(y^L + rand * (y^U - y^L)) \quad (9)$$

式中, $x^L, x^U; y^L, y^U$ 分表表示 x, y 的下界和上界. 在初始种群时,先找一个可行解,其他随机生成,这样能够保证算法中至少有一个正确搜索方向,又使得在初始可行解上的时间最少.

3.2 自适应学习步长

蝙蝠算法根据平均响度对最优位置进行随机扰动(式(4)),由于响度 A_i^t 有着更新条件的限制,特别是到算法早熟收敛时,响度 A_i^t 很难再得到更新,也就是得响度 A_i^t 无法像预期那样随着迭代的进行而趋向零,进而使得对最优位置 X_* 的扰动项 ϵA^t 在后期过大,无法在最优位置 X_* 附近进行精确搜索,做了无用功;为了避免这种情况,提出了自适应学习步长,当 X_{new} 优于 X_i^t , 则 $(X_{\text{new}} - X_i^t)$ 是一个使得 X_i^t 飞向更好位置 X_{new} 的步长(速度),如果只将 $(X_{\text{new}} - X_i^t)$ 作为下代的学习步长,一旦 $(X_{\text{new}} - X_i^t)$ 步长过小,则会导致对 X_{new} 的扰动精度过早降低,无法达到进化效果,所以将 $(X_{\text{new}} - X_i^t)$ 与上代的步长 b_i^{t-1} 进行随机组合成一个新的步长;这样就能避免扰动精度过大或者过早降低;当 X_{new} 不优于 X_i^t 时,说明步长过大,则需要对步长进行适当压缩:

$$X_{\text{new}} = X_* + \epsilon A^t * b_i^t \quad (10)$$

$$b_i^t = \begin{cases} \phi b_i^{t-1} + (1 - \phi)(X_{\text{new}} - X_i^t), & X_{\text{new}} \text{ 优于 } X_i^t \\ 0.95b_i^t, & \text{否则} \end{cases} \quad (11)$$

针对整数变量的局部搜索,因为与整型位置最近的距离都是单位 1, 所以本文采用了单位步长的搜索策略,随机前进,或后退一个单位,或不变:

$$X_{\text{new}}(j) = X_{\text{new}}(j) + ceil(-2 + 3rand) \quad (12)$$

式中, $ceil$ 表示向上取整.

3.3 自然选择^[9]

随着迭代的进行,蝙蝠群可能聚集到局部极值

附近,种群多样性降低,不利于种群跳出局部极值,出现过早收敛.种群多样性决定了全局搜索能力,为了保证种群多样性,当算法更新完所有粒子的位置后,根据粒子与当前最优位置的距离,在算法中将离当前最优位置最远的部分粒子舍去,同时在搜索空间随机产生等量新粒子.若舍去的粒子数量越大,则产生的新粒子越多,有利于保持种群的多样性,避免算法陷入局部最优.如果过大,则会使算法趋于随机搜索;随机产生的数量越少,则越不利于算法维持粒子的个体多样性,算法的全部搜索能力变弱.因此这里取种群中最差的 $15\% \times popsize$ 进行随机产生,其中 $popsize$ 为种群规模.

3.4 违反约束度和约束条件处理

处理问题(1)中等式约束通常转化成不等式约束,即 $|h_j| - \epsilon \leq 0$,其中 ϵ 为等式约束的容忍度,一般取较小的正数,这样就统一成小于等式约束了,采用式(13)计算每个约束条件的违法度:

$$F_i(X) = \begin{cases} 0, & g_i(X) \leq 0 \\ |g_i(X)|, & g_i(X) > 0 \end{cases} \quad (13)$$

同时个体 X 总的约束违反度取:

$$f_{con}(X) = \sum_{i=1}^{m+l} \frac{w_i}{\sum(w)} F_i(X) \quad (14)$$

式中, w_i 属于 $[0, 1]$ 随机获得的, $w = [w_1, w_2, \dots, w_{m+l}]$, m 为不等式约束条件的数量, l 为等式约束条件的数量.

本文根据如下原则判断两个蝙蝠位置的优劣:

(1) 如果两个位置都是可行解,则适应度值小(本文求解最小值)的蝙蝠更优;

(2) 如果一个位置是可行解,另一个位置是不可行解,则可行解位置更优;

(3) 如果两个位置都是不可行解,则总的约束违反度小的个体更优.

3.5 算法流程

Step1:初始化算法参数,在给定空间中先找到一个可行解,然后随机生成其他种群蝙蝠的位置 X_i ,同时计算相应蝙蝠的函数适应度和违反度,随机产生脉冲发射的速率 R_0 和声音响度 A_0 ,并找出最优位置 X_* .

Step2:蝙蝠 i 根据式(1)~(3)来更新速度和位置,计算适应值和违反度.

Step3:如果随机生成的数 $rand1$ 大于蝙蝠 i 的脉冲速率 r_i^t ,则蝙蝠 i 按式(10)、(11)在最优位置 X_* 附近生成一个局部解 X_{new} ,再按式(12)更新 X_{new} 的整数变量,如果 X_{new} 优于 X_* ,则更新全局最

优位置.

Step4:若随机生成的数 $rand2$ 小于蝙蝠 i 的脉冲响度 A_i^t ,根据判断原则得出 X_{new} 优于 X_* ,则将蝙蝠 i 移向 X_{new} ,并且按式(6)减小蝙蝠 i 的脉冲响度 A_i^t ,按式(7)增大蝙蝠 i 的脉冲速率 r_i^t .

Step5:对种群按与当前最优位置的距离大小进行排序,对最远的一部分蝙蝠(15%),按式(8)、(9)重新随机生成位置,更新函数的适应度和违反度.

Step6:根据求解判断终止条件是否成立,若成立,输出全局最优位置 X_* 及 $f(X_*)$;否则转至步骤 Step2.

4 仿真实验与分析

为了检验改进的合理性和有效性,对来自文献[10]的13个带约束的非线性混合整数规划问题的常用标准测试函数进行试验.在文献[10]改进的六个算法中,与每个函数测试效果最好的两个改进粒子群算法进行比较(表1).并与求解混合整数规划问题的几种经典改进算法^[4](MI-LXPM 算法、RST2ANU 算法、AXNUM 算法)进行成功率和收敛代数比较(表2).改进算法种群规模为40,最大迭代次数 $\text{inmax} = 1\,000$, $f_{\min} = 0$, $f_{\max} = 2$, $\alpha = \gamma = 0.9$, $A_i^0 = R_i^0 = \text{rand}$,对每个问题运行50次.从最优值、最差值、平均值、方差、平均收敛代数和成功率多个方面进行分析比较,算法的成功率是指目标函数值与已知目标函数值的差小于0.1%.

从表1可以看出,对于问题 p1~p12,改进的蝙蝠算法都能在1 000代的进化过程中求出最优解,特别是对最大值问题 p08 目前已知最优解位置是 $[0, 1, 1, 1, 0, 1, 1, 0]$,最大值为0.943 47,而本文算法求得最优位置 $[1, 1, 0, 1, 0, 1, 1, 0]$,最大值为0.953 197,所求解优于目前所知最优解;从求解问题 p01~p03 和 p05~p12 的结果可以看出,达到的精度、方差、平均值在一定程度上都比其他算法所求结果好,而且平均收敛代数都少于 CLSPSO,有的甚至相差好几十倍;对于问题 p04,改进的蝙蝠算法(ANBA)能够找到最优解,但方差和平均收敛代数较差于 CLSPSO 算法,说明在 ANBA 算法稳定性上还稍有不足;对于最大值问题 p13,本文算法求出的最优解好于测试函数给出的最优解,但在1 000代进化所求解比 CLSPSO 算法所求最优解还有差距,通过增加迭代次数需要达到三万多代才会达到1 352 439, p13 的最优位置在 $99 * \text{ones}(1, 40)$,即所有变量都在99取得,说明算法的全局搜索能力在

面对 p13 这样的问题不是很好;对于其他问题,本文
算法均能有效地求解出最好解. 与 CLSPSO 算法相
比,改进的蝙蝠算法对大部分测试函数,都能够有效

提高求解精度和收敛速度,这与蝙蝠算法的搜索习
性是紧密相关的.

表 1 各算法运行结果

测试问题 (已知最优解)	算法	最优值	最差值	平均值	方差	平均收敛代数	成功率/%
p01	ANBA	2	2	2	0	47	100
(2)	CLSPSO	2	2	2	6.22E-14	284	100
	CLSPSO2	2	2.000 1	2	1.90E-05	182	100
p02	ANBA	2.124 4	2.124 4	2.124 4	4.51e-16	44	100
(2.124)	CLSPSO	2.124 5	2.124 5	2.124 5	1.71E-07	248	100
	CLSPSO2	2.124 5	2.124 5	2.124 5	1.38E-05	258	100
p03	ANBA	1.076 54	1.076 68	1.076 54	2.62E-05	216	100
(1.076 54)	CLSPSO	1.076 5	1.089 4	1.077 1	0.002	1 229	86
	CLSPSO2	1.076 6	1.082 8	1.079	0.001 7	1 386	28
p04	ANBA	−4 242.005	−4 242.002	−4 242.004	4.25e-4	678	99
(−4 242.004 73)	CLSPSO	−4 242	−4 242	−4 242	3.67E-12	259	100
	CLSPSO2	−4 242	−4 241.8	−4 242	0.055 7	341	100
p05	ANBA	−68	−68	−68	0	13	100
(−68)	CLSPSO	−68	−68	−68	0	222	100
	CLSPSO3	−68	−68	−68	0	191	100
p06	ANBA	−6	−6	−6	0	11	100
(−6)	CLSPSO	−6	−6	−6	0	38	100
	CLSPSO3	−6	−6	−6	0	39	100
p07	ANBA	32 217.422	32 217.422	32 217.422	1.48E-11	12	100
(32 217.42)	CLSPSO	32 217	32 217	32 217	2.20E-11	588	100
	CLSPSO3	32 217	32 158	3.221	19.648 7	2 908	88
P08	ANBA	0.953 197	0.953 197	0.953 197	3.39E-16	9	100
(0.943 47)	CLSPSO	0.943 5	0.943 5	0.943 5	1.23e-15	1 788	100
	CLSPSO2	0.943 5	0.931 6	0.943 2	0.001 7	1 127	98
P09	ANBA	8	8	8	0	30	100
(8)	CLSPSO	8	8	8	0	66	100
	CLSPSO3	8	8	8	0	65	100
p10	ANBA	−42.632 1	−42.632 1	−42.632 1	7.23E-15	10	100
(−42.632)	CLSPSO	−42.632 1	−42.632 1	−42.632 1	7.18E-15	31	100
	CLSPSO3	−42.632 1	−42.632 1	−42.632 1	7.18E-15	32	100
p11	ANBA	7.70E-34	5.16E-09	4.93E-10	1.26E-09	11	100
(0)	CLSPSO	3.79E-08	3.42E-04	1.19E-05	5.25E-05	385	100
	CLSPSO3	2.34E-07	5.17E-05	5.92E-06	1.05E-05	218	100
p12	ANBA	807	808	807.2	0.379 0	502	99
(807)	CLSPSO	807	819	807.88	2.412 9	7 643	78
	CLSPSO2	807	879	829.98	16.562 2	8 123	4
p13	ANBA	1 352 439	1 352 439	1 352 439	0	32 314	100
(1 352 439)	CLSPSO	1 352 439	1 352 439	1 352 439	0	57 002	100
	CLSPSO1	1 352 439	1 249 959	13 366	28 022	1 916	62

由表 2 可以看出,对于大多数问题,本文算法的成功率为 100%,并且比其他 3 种算法的成功率高;对于问题 p13,由于文献[4]已知最优解是 1 030 361,所以三种算法成功率都达到 100%.

表 2 与三种经典算法的成功率比较 %

测试函数	ANBA/	MI-LXPM/	RST2ANU/	AXNUM/
p01(min)	100	88	43	87
p02(min)	100	80	59	64
p03(min)	100	48	5	40
p04(min)	99	89	2	85
p05(min)	100	100	77	92
p06(min)	100	100	100	100
p07(max)	100	100	100	100
p08(max)	100	90	98	41
p09(min)	100	100	100	95
p10(min)	100	97	100	93
p11(min)	100	100	100	100
p12(min)	99	91	21	7
p13(max)	100	100	100	100

通过与不同算法的各方面结果比较可以看出,改进的蝙蝠算法能够很好求解非线性混合整数规划问题,特别是针对问题 p08,改进的蝙蝠算法求解出比目前已知最优解更好的解,这不仅是改进策略的效用,更是与蝙蝠算法的搜索特点离不开的,蝙蝠算法的发展时间还不长,但是作为一种新型智能算法,是值得在不同问题上进行研究和发展的.

5 结束语

本文提出了一种改进的蝙蝠算法求解混合整数规划问题.改进的蝙蝠给出了一种自适应的学习步长,提高了局部搜索能力;同时采用了自然选择策略增加种群多样性,平衡全局搜索能力.通过实验结果表明,改进的蝙蝠算法能有效且更好的解决大部分问题,但对于最优解位于边界的搜索速度较慢,需要进一步研究.

参考文献:

[1] Westerlund T, Skrifvars H, Harjunkski I, et al. An extended cutting plane method for a class of non-convex MINLP problems[J]. Computers & Chemical En-

gineering, 1998, 22(3): 357-365.
[2] Quesada I, Grossmann I E. An LP/NLP based branch and bound algorithm for convex MINLP optimization problems [J]. Computers & chemical engineering, 1992, 16(10):937-947.
[3] Exler O, Antelo L T, Egea J A, et al. A Tabu search-based algorithm for mixed-integer nonlinear problems and its application to integrated process and control system design[J]. Computers & Chemical Engineering, 2008, 32(8): 1877-1891.
[4] Deep K, Singh K P, Kansal M L, et al. A real coded genetic algorithm for solving integer and mixed integer optimization problems[J]. Applied Mathematics and Computation, 2009, 212(2): 505-518.
[5] Jun W, Yuelin G, Lina Y. An Improved Differential Evolution Algorithm for Mixed Integer Programming Problems[C]//Computational Intelligence and Security (CIS), 2013 9th International Conference on. IEEE, 2013: 31-35.
[6] YANG Xinshe. A new metaheuristic bat-inspired algorithm[C]//Nature Inspired Cooperative Strategies for Optimization. IEEE,2010:65-74.
[7] GANDOMI A H, YANG Xin-she, ALAVI A H, et al. Bat algorithm for constrained optimization tasks [J]. Neural Computing and Applications, 2013 ,22 (6):1239-1255.
[8] FISTER I, RAUTER S, YANG Xin-she, et al. Planning the sports training sessions with the bat algorithm [J]. Neurocomputing, 2015,149(2):993-1002.
[9] 白俊强,尹戈玲,孙智伟. 基于二阶振荡及自然选择的随机权重混合粒子群算法[J]. 控制与决策,2012,27 (10):1459-1470.
[10] TAN Yue, TAN Guan-zheng, DENG Shu-guang. Hybrid particle swarm optimization with chaotic search for solving integer and mixed integer programming problems[J]. Journal of Central South University, 2014, 21 (7):2731-2742.

作者简介:

赵乃刚 男,(1975-),硕士研究生,讲师. 研究方向为数据挖掘. E-mail:2821499953@qq. com.
李 勇 男,(1974-),博士. 研究方向为智能算法及无线网络.