

DOI:10.16356/j.1005-2615.2016.05.019

基于小生境技术的改进引力搜索算法

张明¹ 田娜² 纪志成¹ 王艳¹

(1. 江南大学物联网工程学院物联网技术应用教育部工程研究中心, 无锡, 214122; 2. 江南大学人文学院, 无锡, 214122)

摘要:针对引力搜索算法(Gravitational search algorithm, GSA)开发能力强而探索能力弱的特点,提出一种基于小生境技术的引力搜索算法(Niching behavior based advanced GSA, NAGSA)。首先分析了引力搜索算法的性能,为每个粒子定义质量吸引度和欧式距离吸引度两个属性,根据这两个属性计算出粒子吸引概率,取代原有的质量排序选择法。其次,运用吸引概率和小生境拥挤度技术引导粒子在邻域内搜索,平衡算法的收敛速度和多样性。此外,算法将 k best 的取值按照指数函数递减,进一步提高收敛精度。10 个标准测试函数的仿真结果表明,该算法能有效地提高最优解的精度,加快收敛速度。最后,采用 4 个标准柔性作业车间调度模型,验证了该算法在解决实际问题中的可行性和优越性。

关键词:引力搜索算法;小生境技术;质量吸引度;欧式距离吸引度;吸引概率;柔性车间调度

中图分类号: TP301.6 **文献标志码:** A **文章编号:** 1005-2615(2016)05-0753-08

Niching Behavior Based Advanced Gravitational Search Algorithm

Zhang Ming¹, Tian Na², Ji Zhicheng¹, Wang Yan¹

(1. Engineering Research Center of Internet of Things Technology Applications Ministry of Education,

School of IoT Engineering, Jiangnan University, Wuxi, 214122, China;

2. School of Humanities, Jiangnan University, Wuxi, 214122, China)

Abstract: According to the strong exploitation and poor exploration abilities of gravitational search algorithm(GSA), an niching behavior based advanced GSA (NAGSA) is proposed. After analyzing the performance of GSA, NAGSA defines the mass affinity and Euclidean-distance affinity for each particle, and then each particle affinity probability is calculated according to these two attributes instead of the original sorting mass method. The use of affinity probability and crowding niching behavior guides each particle to search in its neighboring field, thus NAGSA can make a balance between convergence rate and diversity maintaining. Besides, the value of k best decreases according to exponential function, so that the convergence accuracy is improved. Simulations on ten benchmark functions indicate that NAGSA can improve the accuracy of optimum effectively and accelerate the convergence rate apparently. Furthermore, the algorithm is proved to be feasible and advantageous in the simulation of four standard flexible job shop scheduling modules.

Key words: gravitational search algorithm; niching behavior; mass affinity; Euclidean-distance affinity; affinity probability; flexible job shop scheduling

基金项目:国家高技术研究发展计划(“八六三”计划)(2014AA041505)资助项目;国家自然科学基金(61572238)资助项目;江苏省杰出青年基金(BK20160001)资助项目。

收稿日期:2015-01-21; **修订日期:**2015-12-27

通信作者:王艳,女,教授,博士生导师,E-mail:wangyan88@jiangnan.edu.cn。

引用格式:张明,田娜,纪志成,等.基于小生境技术的改进引力搜索算法[J].南京航空航天大学学报,2016,48(5):753-760. Zhang Ming, Tian Na, Ji Zhicheng, et al. Niching behavior based advanced gravitational search algorithm[J]. Journal of Nanjing University of Aeronautics & Astronautics, 2016, 48(5): 753-760.

2009年,伊朗的克曼大学提出了基于牛顿物理学的引力搜索算法^[1](Gravitational search algorithm, GSA),它是一种源于牛顿万有引力定律和质量相互作用理论的群体智能优化方法。它的搜索粒子是质量的集合,即它的群体是一个质量的孤立系。在孤立系中,每个质量可以通过万有引力观察其他质量的状态,因此万有引力是不同质量间交换信息的途径。文献[1]将GSA算法与粒子群算法(Particle swarm optimization, PSO)、中心力优化算法(Central force optimization, CFO)以及实数遗传算法(Real genetic algorithm, RGA)做比较,研究结果表明GSA算法在收敛速度和收敛精度上都体现出明显优势。

GSA算法的不足之处在于容易早熟和收敛停滞,以致陷入局部最优的状态。原因在于所有粒子都向种群中质量大的其他粒子学习,那么随着迭代次数的增加,种群渐渐向局部最优解靠拢,搜索空间变小的同时,多样性减少,算法不可避免地陷入局部最优状态。近年来,为了提高种群多样性和抑制早熟,学者们提出了多种改进的引力搜索算法,如文献[2]在GSA算法中引入了量子行为,文献[3]运用混沌策略对算法进行局部搜索,文献[4]模仿PSO算法保存粒子的历史最优值提高算法的开发能力和收敛精度,而文献[5,6]均是将PSO算法和GSA算法融合。尽管改进算法各有优势,改善GSA算法的性能仍然是难点,如文献[2]改善了开发能力却没有明显提高探索能力。

针对GSA算法善于开发而劣于探索的特点,本文提出一种基于小生境技术^[7]的改进引力搜索算法(Niching behavior based advanced gravitational search algorithm, NAGSA)。小生境技术是指能够搜索并且保存多稳定小生境的技术,或者是指将种群维持在多解的周围,避免种群收敛到单解。小生境技术可以使进化算法的种群维持多样性并且搜索到不同的局部最优解。目前,多种小生境技术被陆续提出,包括拥挤度^[8]、适应值共享^[9]、物种形成^[10]、限制锦标赛选择法^[11]等策略。本文为每个粒子定义质量吸引度和欧式距离吸引度两个属性,并且通过线性公式计算出粒子的吸引概率。运用吸引概率和小生境拥挤度技术,使种群同时在多个局部最优解和全局最优解的周围进行细致搜索;此外,通过分析GSA算法的公式,根据指数递减函数将kbest值有效控制控制在10%~5%之内,使得开发与探索得到平衡。

由于NAGSA对多峰问题的求解效果显著,

因此用该算法解决柔性作业调度问题^[12-14](Flexible job-shop scheduling problem, FJSP)。FJSP包含工序排序和机器分配两个问题。工序排序必须遵守工序先后次序的约束,而机器分配指每道工序可在不同的机器上加工,并且加工时间各不相同。因此FJSP是NP(Non-deterministic polynomial)难问题。算法在4个柔性模型算例上测试取得较好的结果,验证了本算法解决实际问题的可行性和优越性。

1 引力搜索算法

1.1 万有引力法则

牛顿万有引力的作用被称为“在一定距离内的作用”,这意味着两个不同粒子之间的引力是没有延迟和间隔的。根据牛顿万有引力定律,每一个粒子由于万有引力的作用而彼此相互吸引,吸引力的大小与粒子的质量成正比,与他们之间欧式距离的平方成反比

$$F = G \frac{M_1 M_2}{R^2} \quad (1)$$

式中: F 代表万有引力的大小; G 代表引力常数; M_1 和 M_2 分别代表两个粒子的惯性质量; R 为两个粒子之间的欧氏距离。根据牛顿第二定律,当物体受到力 F 的作用时,加速度 a 的大小取决于物体的惯性质量 M 和力 F 的大小

$$a = \frac{F}{M} \quad (2)$$

根据式(1,2),两个粒子的质量越大,距离越短,它们的相互吸引力就越大。图1为力与加速度作用图,图中 F_{ij} 表示 M_j 对 M_i 的作用力, F_i 表示 M_i 的受力总和, a_i 表示 F_i 对 M_i 的加速度。此外由于引力常数取决于宇宙的年龄,引力常数呈递减趋势。主动引力质量 M_a :衡量引力场强度,与引力场强度成正比;被动引力质量 M_b :衡量被引力场影响的强度,在同一个引力场中, M_b 越小,受到引力场的作用越小。惯性质量 M_i :衡量质量在受力时抑制改变当前运动的强度, M_i 越大,越难于改变现状。

考虑到上述方面,质量 M_j 对质量 M_i 的作用力 F_{ij} 与 M_i 的主动引力质量 M_{aj} 和 M_i 的被动引力质量 M_{bi} 的乘积成正比,且与两者之间的距离 R 的平方成反比。而加速度 a_i 与力 F_{ij} 成正比,与惯性质量 M_{bi} 成反比。因此式(1,2)可改写为

$$F_{ij} = G \frac{M_{aj} M_{bi}}{R^2} \quad (3)$$

$$a_i = \frac{F_{ij}}{M_{ii}} \quad (4)$$

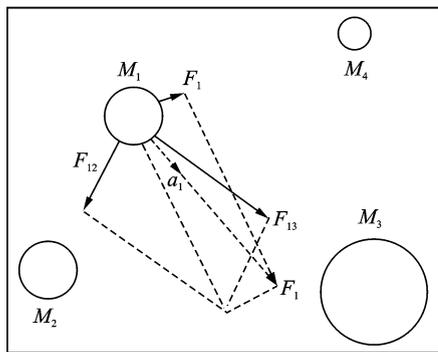


图 1 力与加速度作用图

Fig. 1 Action diagram of force and acceleration

1.2 引力搜索算法介绍

在引力搜索算法中,所有的粒子均受到引力吸引,这就使得种群向质量较大的粒子做全局靠拢。因此引力是粒子之间传递信息的媒介,质量越大的解,移动的越少,这有助于算法的开发能力。GSA 算法有 4 个要素:位置、惯性质量、主动引力质量和被动引力质量。质量的位置表示问题的解,惯性质量和引力质量由适应值决定。

假设在一个 D 维的搜索空间中有 N 个粒子,定义第 i 个粒子的位置为

$$X_i = (X_i^1, X_i^2, \dots, X_i^D) \quad i=1,2,\dots,N \quad (5)$$

式中: X_i^d 表示粒子 X_i 的第 d 维。

在 t 次迭代,定义粒子 j 对粒子 i 作用力 $F_{ij}^d(t)$ 为

$$F_{ij}^d(t) = G(t) \frac{M_{bi}(t) \times M_{bj}(t)}{R_{ij}(t) + \epsilon} (X_j^d(t) - X_i^d(t)) \quad (6)$$

式中: $G(t)$ 表示 t 时刻的引力常数(式(7)); ϵ 为很小的常数; R_{ij} 表示粒子之间的欧式距离(式(8))

$$G(t) = G_0 \exp(-\beta * t/t_{\max}) \quad (7)$$

式中: G_0 设为 100, β 为 20(参考文献[1]); t_{\max} 为最大迭代次数。

$$R_{ij}(t) = \| X_i(t), X_j(t) \|_2 \quad (8)$$

为了增加随机性,假设粒子 X_i 的第 d 维的总作用力为其他所有粒子的作用力之和, X_i 的受力和加速度定义分别为

$$F_i^d(t) = \sum_{j=1, j \neq i}^{k_{\text{best}}} \text{rand}_j F_{ij}^d(t) \quad (9)$$

式中: rand_j 为 $[0,1]$ 之间的随机数。

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)} \quad (10)$$

那么,粒子 X_i 的更新速度为部分当前速度与加速

度之和,因此位置和速度更新公式为

$$v_i^d(t+1) = \text{rand}_i \times v_i^d(t) + a_i^d(t) \quad (11)$$

$$X_i^d(t+1) = X_i^d(t) + v_i^d(t+1) \quad (12)$$

式中: rand_i 为 $[0,1]$ 之间的随机数,它可以增加搜索的随机性。

物体的惯性质量依据其适应度值的大小来计算,惯性质量越大表明它越接近最优值,同时意味着该物体的吸引力越大,但其移动速度却越慢。假设引力质量与惯性质量相等,物体的质量可以通过适当的运算规则去更新,更新算法为

$$M_{ai} = M_{bi} = M_{ii} = M_i \quad i=1,2,\dots,N \quad (13)$$

$$m_i(t) = m_i(t) = \frac{(\text{fit}_i(t) - \text{worst}(t))}{(\text{best}(t) - \text{worst}(t))} \quad (14)$$

$$M_i(t) = m_i(t) / \sum_{j=1}^N m_j(t) \quad (15)$$

式中: $\text{fit}_i(t)$ 表示第 t 次迭代时 X_i 的适应值;针对极小化问题时, $\text{worst}(t)$ 和 $\text{best}(t)$ 的定义见式(16),极大化问题见式(17)

$$\begin{cases} \text{best}(t) = \min_{j \in \{1,\dots,N\}} \text{fit}_j(t) \\ \text{worst}(t) = \max_{j \in \{1,\dots,N\}} \text{fit}_j(t) \end{cases} \quad (16)$$

$$\begin{cases} \text{best}(t) = \max_{j \in \{1,\dots,N\}} \text{fit}_j(t) \\ \text{worst}(t) = \min_{j \in \{1,\dots,N\}} \text{fit}_j(t) \end{cases} \quad (17)$$

为了避免陷入局部最优,算法在初期必须处于探索阶段,随着迭代次数的增加,探索能力渐渐减弱,而开发能力逐步增强。GSA 通过控制开发和探索能力提高收敛速度,因此迭代过程中,只有 k_{best} 个质量较好的粒子对种群中的其他粒子产生引力, k_{best} 的初始值为种群数量,并随着迭代次数的增加而线性减少,因此粒子 X_i 的受力公式为

$$F_i^d(t) = \sum_{j=k_{\text{best}}, j \neq i} \text{rand}_j F_{ij}^d(t) \quad (18)$$

1.3 引力搜索算法性能分析

在引力搜索算法中,每个粒子通过万有引力吸引其他粒子,粒子间的吸引力与质量和欧式距离有关。假设种群处于初期阶段($k_{\text{best}} = N$,换言之,所有粒子都对当前粒子有吸引力)。图 2 为引力搜索算法粒子移动选择图,图中 X 为当前粒子,种群被划为 4 个区域: A, B, C, D 。粒子 X 运用式(6, 10)计算引力和加速度后更新速度时,有以下推断:

(1) 4 个区域中的粒子对 X 的吸引力相差不大。以 A 区为例,虽然质量最大,但是因为距离最远,吸引力变小;同样 B 区中的粒子虽然质量最差,但是因为距离最近,吸引力反而变大,因此 X 的更新充满随机性。

(2) 对于 A 区的粒子而言,应该在本区域做进

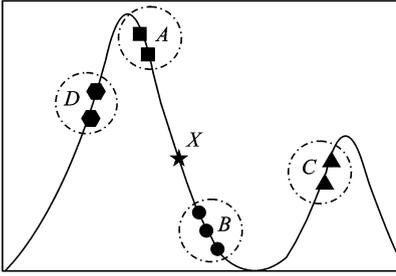


图2 引力搜索算法粒子移动选择图

Fig. 2 Particle selected moving of GSA

一步开发,以便搜索到极值点,然而如果B区的粒子足够多的话,A区的粒子会向B区移动,这反而降低了精度。

从以上分析可以看出:

(1)在迭代初期,当所有的粒子都有引力时,种群更新不合理,也就是说, k_{best} 的取值是不合理的,这会降低种群的开发能力。

(2)粒子之间的欧式距离对粒子的更新是否有效起到很大的作用,应该提高欧式距离影响力。

(3)种群采用质量排序法选择粒子,在迭代后期种群容易陷入局部最优。

以 Sphere 函数(单峰)和 Griewank 函数(多峰)为例(设种群大小为 60,维度为 30,迭代次数为 500),将 GSA 算法、人工蜂群算法(Artificial bee colony, ABC)和量子粒子群算法(Quantum particle swarm optimization, QPSO)进行比较,收敛曲线见图 3,4。通过对比可以看出,在 Sphere 函数中,GSA 算法的收敛精度和收敛速度都优于 ABC 算法和 QPSO 算法;而对于 Griewank 函数,GSA 算法陷入早熟。因此,GSA 算法在单峰问题上表现出很好的开发能力,然而探索多峰问题的能力较差,证明了上述推断。

2 基于小生境技术的改进引力搜索算法

从第 1.3 节的分析可以看出,GSA 算法需要改进两点:(1)引入吸引概率;(2)有效控制 k_{best} 取值。

2.1 改进的引力搜索算法

(1)引入吸引概率

不同于 GSA 算法采用质量排序法,本文同时考虑质量和欧式距离两个要素,为每个粒子赋予质量吸引度 MA 和距离吸引度 EA,则

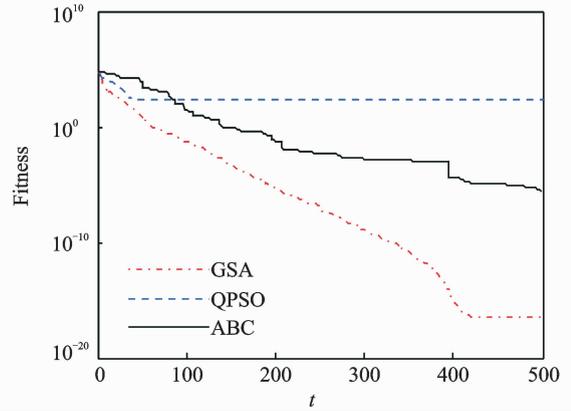


图3 Sphere 函数收敛曲线

Fig. 3 Convergence of Sphere function

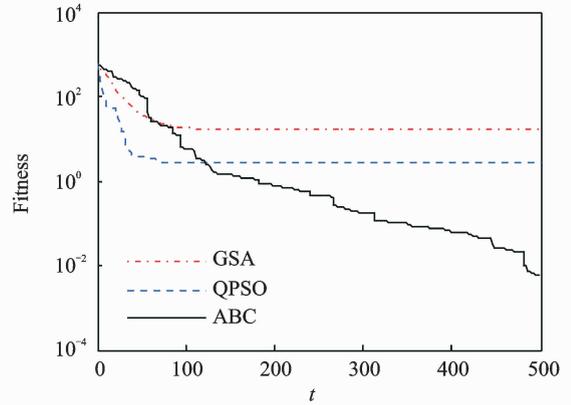


图4 Griewank 函数收敛曲线

Fig. 4 Convergence of Griewank function

$$MA_{ij} = \frac{\exp(0.1 \times (M_j - M_i))}{\sum_{j=1}^N \exp(0.1 \times (M_j - M_i))} \quad (19)$$

$$EA_{ij} = 1 - \frac{R_{ij}}{\sum_{j=1}^N R_{ij}} \quad (20)$$

式中: MA_{ij} 和 EA_{ij} 分别为 j 粒子对 i 粒子的质量吸引度和距离吸引度; R_{ij} 表示两者之间的欧式距离。 j 粒子对 i 粒子的吸引概率 AP 为

$$AP_{ij} = \tau \times EA_{ij} + (1 - \tau) \times MA_{ij} \quad (21)$$

式中: τ 为选择参数, τ 值越大表示对欧式距离的依赖性越强, τ 越小表示对质量的依赖性越强。本文取 τ 为 0.7, 以欧式距离的影响为主, 质量影响为辅。 AP_{ij} 越大, 被选中的概率越高。通过 AP 选择吸引粒子, 对种群的开发和探索能力起到平衡的作用。

(2)改进的 k_{best} 取值

k_{best}^m 根据指数函数递减, 见式(22)(其中 $\lceil \cdot \rceil$ 符号为向上取整函数), 递减区间为

[10%, 5%], 在迭代初期, $kbest^m$ 取值较大, 有利于保持多样性, 之后 $kbest^m$ 快速缩小能提高收敛精度。图 5 给出了 $kbest^m$ 的取值百分比曲线。

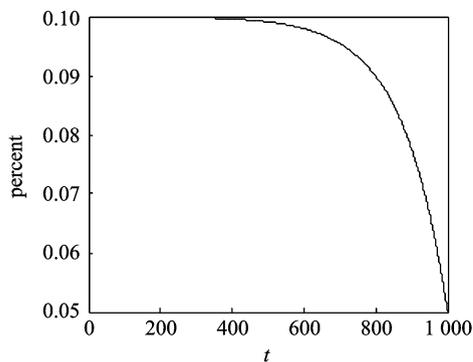


图 5 $kbest^m$ 的取值百分比

Fig. 5 Percent of $kbest^m$

$$kbest^m = \left[N \times \left(10 - 5 \times \left(\frac{\exp(8 * t/t_{max}) - 1}{\exp(8) - 1} \right) \right) / 100 \right] \quad (22)$$

因此, 式(9)改为

$$F_i^d(t) = \sum_{j=1, j \neq i}^{kbest^m} rand_j F_{ij}^d(t) \quad (23)$$

2.2 改进的引力搜索算法

2.2.1 小生境技术

本文采用拥挤度小生境技术, 使进化算法的种群维持多样性并且搜索到不同的局部最优解。拥挤度技术是一种最简单的小生境技术, 它的思想源于对自然种群中的有限数量的相似个体进行竞争(相似度通常用欧式距离衡量)。拥挤度技术能够保持种群的多样性, 它需要将更新后的粒子与从种群中随机选择的子群体进行对比, 因此它需要用拥挤度因子(Crowding factor, CF)对对比个体的数量进行控制, 本文中 CF 等于种群的数量。

为了保持种群的稳定性, 本算法在每次迭代过程中, 将更新得到的新粒子暂存在记忆矩阵 M_x 中, 在本次迭代完成后, 找出种群中与 M_x 最相似的粒子, 运用贪婪算法进行选择^[12]。

2.2.2 NAGSA 算法伪代码

算法的初始条件设置完成后, 算法可概括为以下步骤:

步骤 1 生成初始种群 $X_i = (X_i^1, \dots, X_i^d, \dots, X_i^D)$, $i=1, 2, \dots, N$ 。

步骤 2 在每一次的迭代过程中:

①根据式(21, 22)分别求出 AP, $kbest$;

②对种群中的每个粒子而言, 首先运用式(23, 10)计算粒子的加速度; 其次运用式(11)计算粒子

的速度; 最后运用式(12)得到更新粒子 $X_i(t+1)$;

③将 $X_i(t+1)$ 保存到记忆矩阵 M_x 中;

④对于记忆矩阵 M_x 中的每一个粒子 m_x : 根据欧式距离找出种群中与其最相似的粒子 X_i^{nbest} : $X_i^{nbest} \leftarrow \min_{k \in \{1, \dots, CF\}} R_{ik}$, 如果 m_x 的适应值大于 X_i^{nbest} , 那么 X_i^{nbest} 将被 m_x 替换; 否则, 保留 X_i^{nbest} ;

步骤 3 判断是否满足迭代结束条件, 如果满足则输出全局最优解, 否则返回(2)。

算法的流程图如图 6 所示。

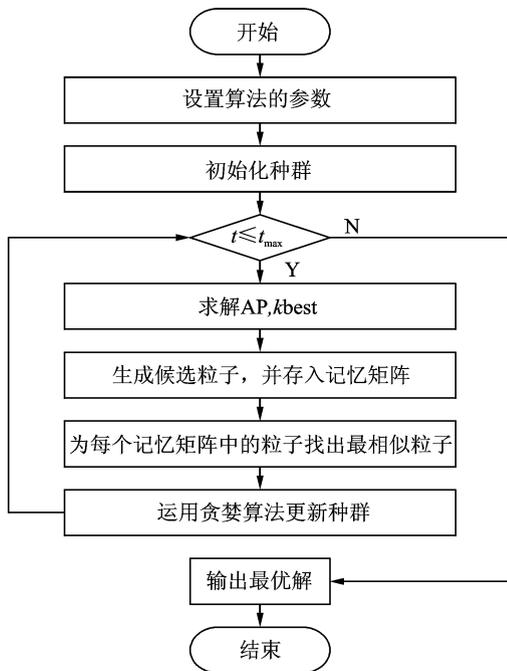


图 6 NAGSA 算法流程图

Fig. 6 Flowchart of NAGSA

3 实验结果与分析

实验种群规模 SN 设为 75, 维度 D 为 30, 迭代次数 t 为 2 000(最大函数评估次数为 150 000), 运行次数 N_{run} 为 30。NAGSA 算法在 10 个 benchmark 函数上进行测试, 结果见表 1。 $f_1 \sim f_5$ 为连续单峰模型函数, f_6 为步进函数。 $f_7 \sim f_{10}$ 为多峰模型函数, 且多峰局部极值点随着维度的增加成指数增长。为测试 NAGSA 算法的性能, 本文将 NAGSA 算法、GSA、HPSO-GSA^[5] 和 BPSOG-SA^[15] 仿真结果进行比较, 表 2 记录了 4 种算法在 10 个函数下的均值(Mean)和方差(S. D.)。从表 2 中的数据可知, 在处理单峰问题时, 该算法对函数 f_1, f_2, f_3, f_5 的优化效果明显优于其他算法, 尤其是对函数 f_1 和 f_3 , 收敛精度有较大幅度的提高, 只是对函数 f_4 的精度略低于 HPSO-GSA。图 7~10 给出了函数 f_1, f_2, f_8, f_9 在不同算法下的

表1 10个标准测试函数

Tab.1 Ten standard test functions

函数	取值范围	最小值
$f_1(X) = \sum_{i=1}^D x_i^2$	$[-100, 100]^D$	0
$f_2(X) = \sum_{i=1}^D (10^6)^{(i-1)/(D-1)} x_i^2$	$[-100, 100]^D$	0
$f_3(X) = \sum_{i=1}^D ix_i^2$	$[-10, 10]^D$	0
$f_4(X) = \sum_{i=1}^D x_i ^{(i+1)}$	$[-10, 10]^D$	0
$f_5(X) = \sum_{i=1}^D x_i^2 + \prod_{i=1}^D x_i $	$[-10, 10]^D$	0
$f_6(X) = \sum_{i=1}^D (x_i + 0.5)^2$	$[-100, 100]^D$	0
$f_7(X) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-10, 10]^D$	0
$f_8(X) = \sum_{i=1}^D [y_i^2 - 10\cos(2\pi y_i) + 10]$	$[-5.12, 5.12]^D$	0
$f_9(X) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]^D$	0
$f_{10}(X) = -20\exp\left(-0.2\sqrt{1/D \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 2 + e$	$[-500, 500]^D$	0

收敛进化曲线。在图7,8中NAGSA的收敛进化曲线稳定递减,这表明该算法在处理单峰问题时,能很好地平衡开发和探索能力,保证收敛速度的同时,不失多样性和稳定性。值得注意的是,4个算

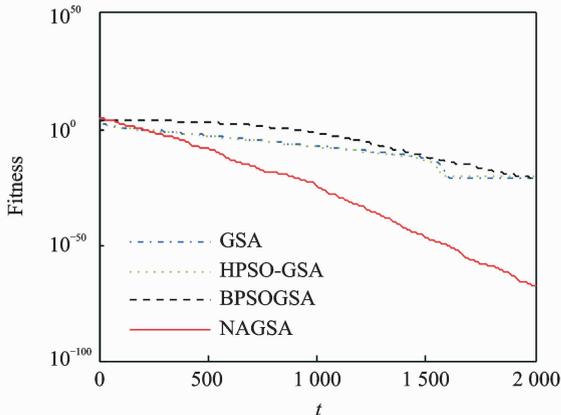


图7 函数 f_1 收敛进化曲线

Fig.7 Convergence of function f_1

表2 实验数据比较

Tab.2 Comparison of experimental data

function		GSA	HPSO-GSA	BPSOGSA	NAGSA
f_1	Mean	3.86e-22	3.34e-22	1.82e-22	1.30e-68
	S. D	2.29e-43	1.53e-41	9.12e-43	3.38e-136
f_2	Mean	7.32e+03	7.16e-21	1.52e-17	1.71e-59
	S. D	5.38e+06	8.16e-41	4.25e-34	1.97e-118
f_3	Mean	5.46e-21	1.08e-21	9.49e-24	2.70e-72
	S. D	4.51e-41	6.69e-41	4.35e-47	3.31e-144
f_4	Mean	1.84e-26	2.51e-62	1.56e-50	9.79e-58
	S. D	6.65e-52	3.43e-124	4.39e-100	1.91e-114
f_5	Mean	1.36e-10	1.19e-09	2.45e-11	1.40e-39
	S. D	7.04e-22	1.03e-18	1.55e-22	3.73e-78
f_6	Mean	0	0	0	0
	S. D	0	0	0	0
f_7	Mean	24.32	21.98	34.09	18.96
	S. D	0.022	0.024	6.99e+02	19.9
f_8	Mean	13.26	11.60	18.57	0
	S. D	4.17	1.53	6.82	0
f_9	Mean	0.011	0.034	0	0
	S. D	2.63e-04	5.14e-04	0	0
f_{10}	Mean	4.51e-12	3.49e-11	9.22e-11	8.58e-15
	S. D	3.67e-24	7.89e-22	1.38e-20	1.12e-29

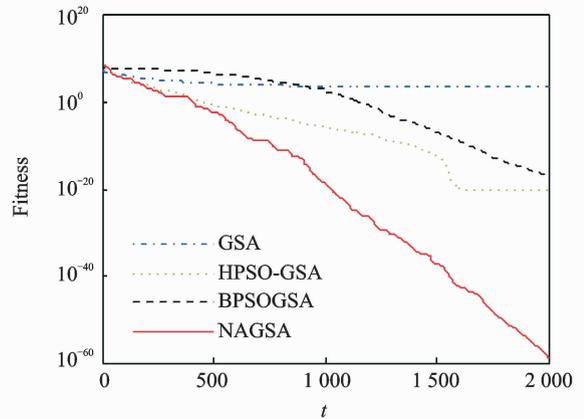


图8 函数 f_2 收敛进化曲线

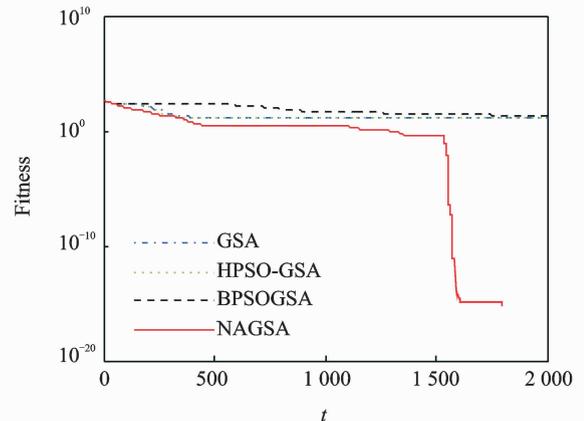


图9 函数 f_8 收敛进化曲线

Fig.9 Convergence of function f_8

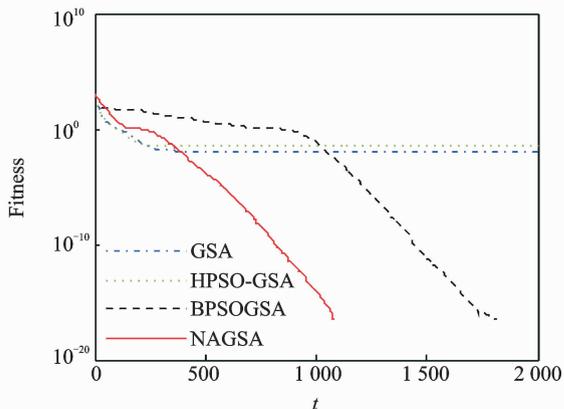


图 10 函数 f_9 收敛进化曲线
Fig. 10 Convergence of function f_9

法对函数 f_6 的优化都能取得最小值,表明引力搜索算法能很好地处理步进函数。

在处理多峰问题时,对于函数 f_7 和 f_{10} , NAGSA 优化效果略高于其他算法,尤其是对 f_7 的效果不甚明显,然而对函数 f_8 和 f_9 的优化却能进化到全局最小值,优化效果明显提高。在图 9 中, NAGSA 在迭代初期的收敛效果不明显,然而在迭代后期却快速收敛,这表明,算法在初期具有很强的探索能力,能够搜索全局找出多个极值峰,而在后期,算法拥有很强的开发能力,能够对潜力峰进行深度优化,进而找出全局最优解。NAGSA 和 BPSOGSA 算法都能将函数 f_9 优化到最小值。从图 10 可以看出, NAGSA 的收敛速度比 BPSOGSA 快,且 NAGSA 的进化曲线更加稳定。

综上所述, NAGSA 能够很好地处理数值优化问题,在单峰和多峰问题上都取得了较好的效果,能够在提高收敛精度和加快收敛速度的同时,确保种群的多样性。

4 NAGSA 算法在柔性作业车间调度问题中的应用

FJSP 问题是组合优化问题,其适应值地形是非常复杂的,包括大山谷、多峰值、或者大山谷和多峰值并存等。因此,群智能算法解决 FJSP 问题,必须考虑算法的多峰优化能力。多峰优化问题主要致力于两部分内容,一方面需要搜索空间中的多个全局和局部峰值点,另外需要在保存各峰值点信息的同时,对各个峰进行进一步搜索,以便找到更好的解。因此,如果在迭代过程中,能够保存峰值点,那么迭代进程结束时,将会得到多个最优解,而不是单峰问题中的一个解,这与 NAGSA 的迭代

过程相同。

4.1 编码和解码

为了检验 NAGSA 算法在实际问题中的有效性,本文将 NAGSA 算法运用到 FJSP 问题中,目标函数为最大完工时间。此时需要考虑到算法的参数 AP, AP 是质量吸引度 MA 和欧式距离吸引度 EA 的线性相加。根据式(14, 15, 19)可求出质量吸引度 MA; FJSP 问题采用文献[16]提供的编码和解码方式,运用每个解编码的位置计算欧式距离吸引度 EA,进而可以求出吸引度 AP。

4.2 仿真结果

采用 MATLAB2013a 开发环境,在标准测试模型中采用 4 个算例进行测试[12]。表 3 为比较结果(其中 Opt 表示算法的最优结果),种群规模为 100,迭代次数为 50,给出了均衡蜂群算法(FER-ABC)[17]、GSA 和 NAGSA 三种算法的比较。其中 3 种算法的种群规模均为 100,迭代次数分别为 100, 100, 50。从对比结果上看, NAGSA 算法的效果优于 GSA 算法,并且在 15×10 模型上的性能比 FER-ABC 算法优越。但是, NAGSA 算法的迭代次数是 FER-ABC 算法的两倍,因此算法的复杂度较高。图 11 给出了 NAGSA 算法 15×10 模型上最优解的甘特图。

表 3 实验结果对比

Tab. 3 Comparison of experimental result

算例	Opt		
	FER-ABC	GSA	NAGSA
4×5	11	11	11
8×8	14	14	14
10×10	7	8	7
15×10	12	12	11

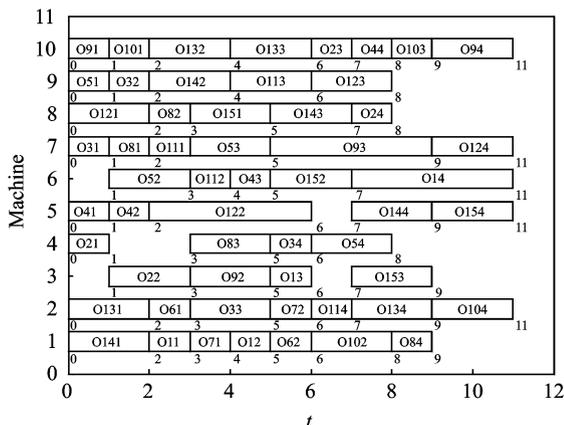


图 11 15×10 模型的甘特图

Fig. 11 Gantt of 15×10 module

5 结束语

本文分析了GSA算法的性能和优缺点,提出一种基于小生境技术的改进引力搜索算法。为每个粒子定义了引力吸引度和欧式距离吸引度属性,并且基于两个属性采用线性叠加的方法定义了吸引概率,替换原有的质量排序法进行粒子选择。此外对 k best取值范围做了改进。在10个benchmark函数上的优化结果表明,该算法表现出很大的优势,收敛速度和收敛精度都有明显提高,能够做到探索和开发的均衡。本文将该算法用于求解柔性作业车间调度问题,在与其他算法的比较中体现其优势。

参考文献:

- [1] Rashedi E, Nezamabadi-Pour H, Saryazdi S. GSA: A gravitational search algorithm[J]. *Information Sciences*, 2009, 179(13): 2232-2248.
- [2] Soleimanpour-Moghadam M, Nezamabadi-Pour H, Farsangi M. A quantum inspired gravitational search algorithm for numerical function optimization [J]. *Information Sciences*, 2014, 267(5): 83-100.
- [3] Gao S, Vairappan C, Wang Y, et al. Gravitational search algorithm combined with chaos for unconstrained numerical optimization[J]. *Applied Mathematics and Computation*, 2014, 231(11): 48-62.
- [4] Jiang S H, Wang Y, Ji Z C. Convergence analysis and performance of an improved gravitational search algorithm[J]. *Applied Soft Computing*, 2014, 24: 363-384.
- [5] Jiang S H, Ji Z C, Shen Y X. A novel hybrid particle swarm optimization and gravitational search algorithm for solving economic emission load dispatch problems with various practical constraints[J]. *International Journal of Electrical Power & Energy Systems*, 2014, 55(2): 628-644.
- [6] Mirjalili S, Wang G G, Coelho L. Binary optimization using hybrid particle swarm optimization and gravitational search algorithm[J]. *Neural Computing and Applications*, 2014, 25(6): 1423-1435.
- [7] Mahfoud S W. Niching methods for genetic algorithms [D]. Urbana: University Illinois Urbana-Champaign, 1995.
- [8] Thomsen R. Multimodal optimization using crowding-based differential evolution[C]// Proceedings of the IEEE 2004 Congress on Evolutionary Computation. Portland: IEEE Xplore, 2004. 1382-1389.
- [9] Kim H, Liou M S. New fitness sharing approach for multi-objective genetic algorithms [J]. *Journal of Global Optimization*, 2013, 55(3): 579-595.
- [10] Chevin L M, Decorzent G, Lenormand T. Niche dimensionality and the genetics of ecological speciation [J]. *Evolution*, 2014, 68(5): 1244-1256.
- [11] Harik G R. Finding multimodal solutions using restricted tournament selection[C]//Proceedings of 6th International Conference of Genetic Algorithms. San Francisco, CA: [s. n.], 1995: 24-31.
- [12] Brandimarte P. Routing and scheduling in a flexible job shop by tabu search[J]. *Annals of Operations Research*, 1993, 41(3): 157-183.
- [13] 高亮, 张国辉, 王晓娟. 柔性作业车间调度智能算法及其应用 [M]. 武汉: 华中科技大学出版社, 2012: 3-26.
Gao Liang, Zhang Guohui, Wang Xiaojuan. Intelligent algorithm and application of flexible job shop scheduling [M]. Wuhan: Huazhong University of Science And Technology Press, 2012: 3-26.
- [14] Li J Q, Pan Q K, Chen J. A hybrid Pareto-based local search algorithm for multi-objective flexible job shop scheduling problems [J]. *International Journal of Production Research*, 2012, 50(4): 1063-1073.
- [15] Biswas S, Kundu S, Das S, et al. Inducing niching behavior in differential evolution through local information sharing[J]. *IEEE Transactions on Evolutionary Computation*, 2014, 19(2): 1-16.
- [16] 白俊杰, 王宁生, 唐敦兵. 一种求解多目标柔性作业车间调度的改进粒子群算法[J]. *南京航空航天大学学报*, 2010, 42(4): 447-453.
Bai Junsheng, Wang Ningsheng, Tang Dunbing. Improved PSO algorithm for multi-objective optimization flexible job shop scheduling problems[J]. *Journal of Nanjing University of Aeronautics & Astronautics*, 2010, 42(4): 447-453.
- [17] 张明, 田娜, 纪志成. 基于适应值欧式距离比的均衡蜂群算法[J]. *系统仿真学报*, 2015, 27(5): 980-989.
Zhang Ming, Tian Na, Ji Zhicheng. Balanced bee colony algorithm based on fitness Euclidean-distance ratio[J]. *Journal of System Simulation*, 2015, 27(5): 980-989.

