

面向多目标优化问题的自适应差分进化算法

刘红平¹ 黎福海²

¹(长沙师范学院电子信息工程系 湖南长沙 410100)

²(湖南大学电气与信息工程学院 湖南长沙 410082)

摘要 针对多目标优化得到一个最优解集和解之间难以比较的问题,对单目标优化中的自适应策略进行了改进,提出一种面向多目标优化问题的自适应差分进化算法,在已有方法自适应改变交叉率的基础上,设定缩放因子有三种不同的分布模型,通过统计一定代数内个体的优劣来自适应选择合适的模型并生成相应取值,从而控制了搜索长度,防止新个体陷入在最优解集的部分区域。该算法还提出利用第三方解集和优胜累积量的概念来处理最优解之间的比较问题。通过 5 个标准优化问题的测试结果以及与其他几种算法的对比研究表明,所提出的改进算法性能更好,其在 IGD 指标上减小了 0.0031~0.0669,在 IH 指标上最多减小了 0.0821。

关键词 多目标优化 差分进化 自适应 缩放因子 优胜累积量

中图分类号 TP391.41 文献标识码 A DOI:10.3969/j.issn.1000-386x.2015.12.058

SELF-ADAPTIVE DIFFERENTIAL EVOLUTION ALGORITHM FOR MULTI-OBJECTIVE OPTIMISATION

Liu Hongping¹ Li Fuhai²

¹(Electronics Information Engineering Department, Changsha Normal College, Changsha 410100, Hunan, China)

²(Electrical and Information Engineering College, Hunan University, Changsha 410082, Hunan, China)

Abstract It is hard to compare an optimal solution set with solutions both derived from multi-objective optimisation, for this problem, we proposed a self-adaptive differential evolution algorithm for multi-objective optimisation, and based on existing algorithm which adaptively changes the crossover rate, we set that the scaling factors have three different distribution models, by counting the advantage and disadvantage of individuals within certain algebra the algorithm adaptively selects proper model and generates corresponding assigned value, so that it controls the search size as well as prevents the new individual falling into partial area of the optimal solution set. This algorithm also puts forward to use the solution set of the third party and the concept of accumulative superiority to deal with the problem of comparison between optimal solutions. It is demonstrated through five standard optimisation problems and the contrast study with some other algorithms that the improved algorithm proposed has better performance, where the IGD index values reduces 0.0031 to 0.0669, and the IH index value reduces up to 0.0821.

Keywords Multi-objective optimisation Differential evolution Self-adaptive Scaling factor Accumulative superiority

0 引言

差分进化是进化计算方法的重要成员,具有算子形式简单、优化性能好和搜索效率高等优点,目前在优化和工程应用领域已经展现出优越的性能^[1-3]。在差分进化中,交叉率 CR 和缩放因子 F 是其主要参数,对算法的性能起着重要的作用^[4-6]。由于待优化对象的特性通常难以准确知晓,并且优化过程中对局部和全局搜索强度的要求也往往是变化的,所以预先指定参数取值的方法不利于算法性能提升,越来越多的学者提出了动态或自适应的参数调整方法^[7]。其中典型算法如 Brest 等人^[8]提出的动态调整算法 jDE, Qin 等人^[9]提出的自适应差分进化算法 SaDE。目前大多数自适应差分进化算法是针对单目标优化问题而设计的,在处理多目标优化问题时,只进行了简单的应用和推广。Huang 等人^[10]采用了与 SaDE 相同的方法对交叉率

CR 进行自适应调整,对缩放因子 F 通过指定高斯分布模型来随机生成,然后融入了多目标优化中常见的基于 Pareto 占优机制和拥挤距离的个体优劣比较准则。邱威等人^[11]采用了和 jDE 类似的动态调整策略,在每次生成新个体时,两个参数的取值均以一定概率随机生成,否则将变异操作中所有个体对应的参数取值的均值作为新的参数取值。但是上述方法对多目标优化的特性考虑不足,不利于问题的求解。首先,已有方法对缩放因子 F 的自学习策略研究不够,如 SaDE 中设定其在一个较小取值附近随机变动。但是多目标优化最终得到的是一个宽广分布在空间中的解集,过大的 F 取值不利于收敛和最优解集的均匀性,过小的 F 取值又容易使算法只能得到最优解集的若干子集区域,不利于保持解集的完整性;其次,在多目标优化过程中,两个

收稿日期:2014-06-10。国家自然科学基金项目(51107036);湖南省科技厅科技计划项目(2012FJ3013)。刘红平,副教授,主研领域:自动化与人工智能。黎福海,教授。

个体容易出现难以比较的现象,即互相不能 Pareto 占优,不利于种群的更新。已有方法往往反复计算整个种群的分布密度来解决此问题,但会耗费大量的计算时间,效果也不是很理想。为此,提出一种面向多目标优化问题的自适应差分进化算法,保留了 SaDE 算法自适应学习交叉率 CR 的策略,同时在参数自适应方面,使缩放因子 F 具有大、中和小三种不同的候选分布,分别对应不同的搜索步长,然后通过一定代数内生成个体的优劣情况自适应地调整各种分布被选中的概率,从而在优化过程中动态地平衡局部和全局搜索的强度;在新个体的优劣判别方面,通过与构建的第三方个体集合的比较来实现,同时基于个体在各个目标函数上的提升量与下降量,提出用优胜累积量的概念作为比较的具体度量。通过对 5 个复杂的标准多目标优化问题的测试,验证了所做改进的有效性。

1 多目标优化与差分进化算子

1.1 多目标优化

多目标优化问题包含两个或两个以上的待优化目标函数,并且目标函数之间存在一定的相互制约,所以不像单目标优化那样能得到单一的最优解,而是得到一个最优解集^[12]。多目标优化问题的定义通常如式(1):

$$F(x) = \{f_i(x) \mid i = 1, 2, \dots, m\} \quad x \in S \quad (1)$$

其中, $x = (x_1, x_2, \dots, x_n)$ 表示可行域 S 中的 n 维变量, S 是 n 维实数域的一个子集, $f_i(x)$ 表示第 i 个目标函数,所有的 m 个目标函数构成了目标空间 $F()$ 。

衡量多目标优化所得解的优劣情况的一个重要概念是 Pareto 占优。若有变量 x_1 Pareto 占优变量 x_2 , 则记为 $x_1 < x_2$ 。其含义为: x_1 在 m 个目标函数上的取值均不大于 x_2 在相应目标函数上的取值(假设为最小化优化问题),并且 x_1 至少在任意一个目标函数上的取值严格小于 x_2 在对应目标函数的取值。进一步,若变量 x_0 在整个可行域 S 中,没有任何一个变量能对其 Pareto 占优,则称 x_0 为 Pareto 解。这样的解不止一个,它们共同构成了多目标优化的理论最优解集,即理论 Pareto 解集。而在优化过程中不可能求出无限个变量对应的所有目标函数值,因而优化得到的 Pareto 解集称为约近 Pareto 解集。约近解集和理论解集的贴合程度越高,说明算法的性能就越好。

1.2 差分进化算子

差分进化的变异和交叉算子有多种具体的表达形式,其中 DE/rand/1/bin 是最常见的一种。本文亦针对该算子进行相应的改进。DE/rand/1/bin 中变异算子的表达式如式(2):

$$v = x_{r_1} + F(x_{r_2} - x_{r_3}) \quad (2)$$

其中, r_1, r_2, r_3 表示从种群中随机选择的三个不同的个体, v 表示变异得到的临时个体。

而 DE/rand/1/bin 的交叉算子在变异算子后执行,其表达式如式(3):

$$u_{ij} = \begin{cases} v_j & (\text{rand}() \leq CR \text{ 或者 } j = j_r) \\ x_{ij} & \text{其他} \end{cases} \quad (3)$$

其中,标号 i , 以及标号 r_1, r_2, r_3 要互不相等,用以保证所对应个体各不相同, j_r 表示从变量中随机选取的一维,用以保证新个体 u_i 不同于父代个体 x_i , $\text{rand}()$ 表示 0 和 1 之间的随机数。

此后,计算由式(2)和式(3)生成的新个体 u_i 与父代个体 x_i 的目标函数值,通过比较来决定是否替换父代个体,从而推动种群的进化。可见,交叉率 CR 和缩放因子 F 决定着新个体的取值,而新个体与父代个体的比较过程则决定着种群的分布情况和寻优方向,它们都是算法中的重要环节。

2 SADEMOO 描述

本文提出的算法 SADEMOO 使得交叉率 CR 和缩放因子 F 的调整策略均具有一定的学习性能,它们的学习过程与新个体相对于父代个体的优劣情况有关。而 SADEMOO 采用的优劣判别准则比 Pareto 占优关系更为细致,避免了难以比较情形的出现,并且具有计算时间少的特性。

2.1 缩放因子的自适应策略

为了便于描述,首先给出 SaDE 中自适应调整交叉率 CR 的方法。假设 CR 服从高斯分布 $g(CR_m, \sigma)$, 其中 CR_m 表示该分布的均值, σ 表示方差。算法在每次计算新个体时,均按照此分布生成所需的 CR 取值。但是这一分布中 CR_m 的取值不是一成不变的, SaDE 设置了一个学习周期 LP (由连续若干代数构成), 期间若新个体优于父代个体,则记录相应的 CR 取值。从第 LP 代开始,将所有记录的 CR 值排序,选取其中间值用于更新 CR_m 。同时记录的 CR 值也在不断更新,在第 i 代时 ($i \geq LP$), 只保留第 $i-LP$ 代至第 i 代的记录内容。

由于交叉率 CR 取值反映了待优化问题的变量耦合程度,令所有个体的交叉率服从同一概率分布模型是合理的,并且上述学习过程也会驱使算法逐渐得到最合适的分布模型。但是缩放因子 F 则不可按照上述方法变化。因为多目标优化得到的是一个最优解集,在优化过程中,若父代个体处于个体分布密集的区域,大的 F 取值下得到的 Pareto 占优解利于发现稀疏区域,保证了 Pareto 解集的完整性;若父代个体处于个体分布稀疏的区域,小的 F 取值下得到的 Pareto 占优解利于加强稀疏区域的密度,保证了 Pareto 解集的均匀性。可见在同一代个体的生成中,不同步长的 F 取值往往是同时需要的。

基于上述分析,本文算法设定缩放因子 F 有三种不同的候选高斯分布模型 $g(F_b, \sigma)$ 、 $g(F_m, \sigma)$ 和 $g(F_s, \sigma)$, 其中 $F_b > F_m > F_s$, 分别对应大、中和小三种不同的搜索尺度。初始阶段三种分布被选中的概率相同,即 $P_b = P_m = P_s$ 。在随后与 CR 相同的学习周期 LP 中,记录每一代中每种分布被选择的总次数 s_b, s_m 和 s_s , 以及新个体优于父代个体时的次数 n_b, n_m 和 n_s 。从而可以按照式(4)计算得到在一个学习周期中各种分布对算法起到有效促进作用的比率 R_b, R_m 和 R_s 。再根据计算所得的比率,即可按照式(5)调整各种分布被选中的概率。

$$R_b = \frac{\sum_{t=i+LP}^{t=i+LP} n_b(i)}{\sum_{t=i+LP}^{t=i+LP} s_b(i)} \quad R_m = \frac{\sum_{t=i+LP}^{t=i+LP} n_m(i)}{\sum_{t=i+LP}^{t=i+LP} s_m(i)} \quad (4)$$

$$R_s = \frac{\sum_{t=i+LP}^{t=i+LP} n_s(i)}{\sum_{t=i+LP}^{t=i+LP} s_s(i)}$$

其中, t 表示进化的代数, $n_b(i)$ 、 $n_m(i)$ 和 $n_s(i)$ 分别表示第 i 代中各种分布下新个体占优的次数, $s_b(i)$ 、 $s_m(i)$ 和 $s_s(i)$ 分别表示第 i 代中各种分布下选择的总次数。

$$\begin{aligned} sum &= R_b + R_m + R_s + 3\varepsilon P_b = \frac{(R_b + \varepsilon)}{sum} \\ P_m &= \frac{(R_m + \varepsilon)}{sum} \quad P_s = \frac{(R_s + \varepsilon)}{sum} \end{aligned} \quad (5)$$

其中, ε 为预先设定的最小量, 用于防止某种分布被选中的概率为零。

通过所提自适应策略, 若算法当前所得约近 Pareto 解集的完整性不够, 则该学习过程中大尺度 F 的分布模型被选中的概率会逐渐增大。若约近 Pareto 解集的均匀性不够, 则较小 F 取值的分布模型被选中的概率也会逐渐增大。同时, 三种分布均保持一定的概率被选中, 可有效兼顾到每一代中父代个体出现在解分布的密集区域或稀疏区域等多种情形。

2.2 基于优胜累积量的比较策略

比较策略是判断新个体优劣的标准, 决定着学习过程的统计结果。SADEMOO 算法为了加强个体之间的可比性, 同时避免评估种群分布密度引起的过高计算代价, 提出选取若干已经得到的 Pareto 占优解构成第三方集合 Q , 新个体 u_i 和父代个体 x_i 均分别与集合 Q 进行比较, 计算它们各自的优胜累积量 $Ac(u_i)$ 和 $Ac(x_i)$, 其中取值大的个体获胜。

本文算法设置了一个存档 A , 用于存储当前已经发现的约近 Pareto 占优解。由于存档 A 的大小有限, 其存储和更新规则为基于格子机制的多目标进化算法 PESAI^[13] 所采用的方案:

- 1) 对于一个解 x_0 , 若 A 中某一个体 Pareto 占优解 x_0 , 则舍弃 x_0 ;
- 2) 否则将 x_0 加入存档 A , 此时若 x_0 相对于 A 中某些个体 Pareto 占优, 则将删除这些个体;
- 3) 当 x_0 的加入使总的个体数量超过 A 的最大限度, 则计算 A 中每个个体对应的超格所包含的个体数量, 并将该数量作为该个体的拥挤度;
- 4) 按照拥挤度排序, 优先舍弃拥挤度大的个体, 直至个体总数等于存档 A 的大小。在构建第三方集合 Q 时, 从当前存档 A 中随机选取 k 个个体来构成。候选个体 u_i 或 x_i 与 Q 中第 l 个 ($l=1, 2, \dots, k$) 个体比较时, 按照式(6)计算优胜累积量。

$$\begin{aligned} Ac(x) &= \sum_{l=1}^{l=k} ac(x, Q(l)) \\ ac(x, Q(l)) &= \sum_{j=1, f_j(x) < f_j(Q(l))}^{j=m} \frac{f_j(Q(l)) - f_j(x)}{f_{\max, j} - f_{\min, j}} - \\ &\quad \sum_{j=1, f_j(x) \geq f_j(Q(l))}^{j=m} \frac{f_j(x) - f_j(Q(l))}{f_{\max, j} - f_{\min, j}} \end{aligned} \quad (6)$$

其中, $Q(l)$ 表示集合 Q 中第 l 个个体自变量取值, $f_{\max, j}$ 和 $f_{\min, j}$ 分别表示第 j 维目标函数上当前已发现的最大和最小值。

优胜累积量的含义为: 个体 x 相比于集合 Q 中的个体, 在一些目标函数上的提升量 (即 $f_j(x) < f_j(Q(l))$) 与在另一些目标函数上的退化量 (即 $f_j(x) > f_j(Q(l))$) 之差。该差值说明了个体 x 在促进各个目标函数寻优时所付出的代价, 即在目标函数上引起的退化量 (退化量 ≥ 0)。优胜累积量越大, 说明优化付

出的代价越小, 相应个体越应保留。

2.3 SADEMOO 的流程描述

SADEMOO 的算法流程:

- 1) 随机生成 $|P|$ 个个体构成种群为 P 并计算相应的目标函数值, 设置进化代数 $t=0$, 存档 A 为空, 将每个目标函数的取值区间划分成等宽的 b 个子区间, 从而构成若干超格, 并设置最大进化代数 t_{\max} 以及所需各个参数的取值;
- 2) 依次将 P 中个体按照 2.2 节所示 PESAI 的更新规则更新存档 A ;
- 3) 设置 $i=1$, 利用参数 CR 和参数 F 的概率分布模型生成相应的参数取值, 再利用算子 DE/rand/1/bin 生成第 i 个新个体, 并生成第三方集合 Q , 用于同父代个体的比较, 再依据 2.1 节中的自适应策略记录参数 CR 的取值, 以及参数 F 所对应分布模型的调用次数和优胜次数;
- 4) 令 $i=i+1$, 重复 3) 直到 $i=|P|$, 得到子代种群 P^* , 用 P^* 替换种群 P ;
- 5) 当 $t \geq LP$ 时, 更新交叉率 CR 的分布模型, 更新缩放因子 F 所有候选分布模型的选择概率;
- 6) 若 $t=t_{\max}$, 则停止迭代, 存档 A 中个体即为最终优化结果, 否则继续, 令 $t=t+1$ 。

3 实验与验证

为了检验所提自适应策略对多目标差分进化算法性能的提升效果, 选取 2007 年进化计算年会上用到的 5 个多目标优化问题作为标准测试函数^[14], 分别为 S_ZDT1、S_ZDT2、S_ZDT4、S_ZDT6 和 S_DTLZ2_M3。这些测试函数的变量维数均为 30 维, 其中前四个测试函数的目标个数是 2 个, 第五个测试函数的目标个数是 3 个。同时设置了两个对比算法, 分别为文献[10]提出的多目标自适应差分进化算法 MOSaDE 和基于文献[8]所提 jDE 方法的多目标算法 MOjDE。各种算法的参数设置为: 对于前四个测试函数, 所有算法用于存储约近 Pareto 解集的存档或种群大小为 100, 第五个测试函数为 150; 对于 SADEMOO, 种群 P 大小为 40, 区间划分数 b 为 32, 均值 CR_m 的初始值为 0.5, 方差 σ 为 0.1, 均值 F_b 、 F_m 和 F_s 的初始值分别为 0.8、0.5 和 0.2, 第三方集合 Q 的大小为 20, 学习周期 LP 为 50; 对于 MOSaDE, 其所有设置与文献[10]相同; 对于 MOjDE, 按照文献[5]所述的参数设置, 用 jDE 中的动态调整方法替换 SADEMOO 中的相应策略, 其余保持和 SADEMOO 相同。所有算法的迭代停止条件为函数调用 5×10^5 次, 每种算法均运行 25 次, 取每次运算结果对应的 IGD 指标^[15] 和 IH 指标^[16] 的均值及方差作为算法性能的评价依据。其中 IGD 指标和 IH 指标均用于考察所得约近 Pareto 解集对理论 Pareto 解集的逼近程度, 以及覆盖的完整性和均匀性, 是常用两种的评价指标。它们的取值越接近零, 说明约近 Pareto 解集的品质越好。在 25 次独立测试中, 本文算法在 5 个函数上的测试结果显示, SADEMOO 所得约近 Pareto 解集与理论 Pareto 解集非常相近, 并且覆盖也较为完整和均匀。三种算法在 IGD 指标和 IH 指标上的计算结果如表 1 所示。

表 1 三种算法测试结果在 IGD 和 IH 指标上的对比

函数	指标	SADEMOO	MOSaDE	MOjDE
S_ZDT1	IGD	0.0031	0.0033	0.0125
	IH	0.0202	0.0185	0.0764
S_ZDT2	IGD	0.0064	0.0247	0.0733
	IH	0.0076	0.0466	0.0897
S_ZDT4	IGD	0.0048	0.0052	0.0112
	IH	-0.0014	-0.0011	0.0035
S_ZDT6	IGD	0.0033	0.0056	0.0064
	IH	-0.0029	-0.0038	0.0077
S_DTLZ2_M3	IGD	0.0103	0.0142	0.0175
	IH	-0.0001	-0.0002	-0.0022

表 1 的结果表明:本文算法在所有测试函数上得到的 IGD 指标最小,在 IH 指标上也仅仅在 S_ZDT1 和 S_ZDT4 上略差于 MOSaDE;SADEMOO 与 MOSaDE 的测试结果均好于 MOjDE,说明具有学习能力的自适应策略在多目标优化中能更好地发挥差分算子的性能。

为测试所提出的缩放因子的调整策略对算法性能的影响,构建了 4 种对比算法,记为 SADEMOO1 - SADEMOO4。它们均采用了单一的缩放因子分布模型,其余则保持和 SADEMOO 算法的设置一致。其中前三种算法的缩放因子均服从高斯分布,分别为 $g(F_b, \sigma)$ 、 $g(F_m, \sigma)$ 和 $g(F_s, \sigma)$,第四种的缩放因子服从 0.1 到 0.9 之间的均匀分布。几种算法测试结果所对应的 IGD、IH 均值(绝对值),如图 1 和图 2 所示。

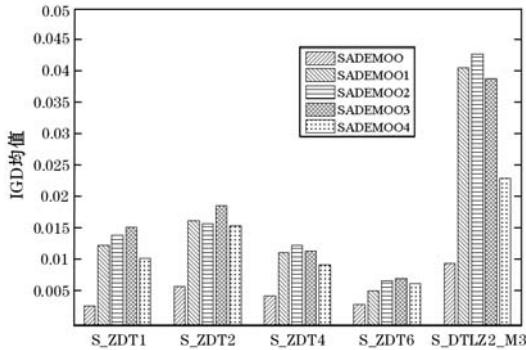


图 1 不同缩放因子调整策略在 IGD 指标上对算法性能影响

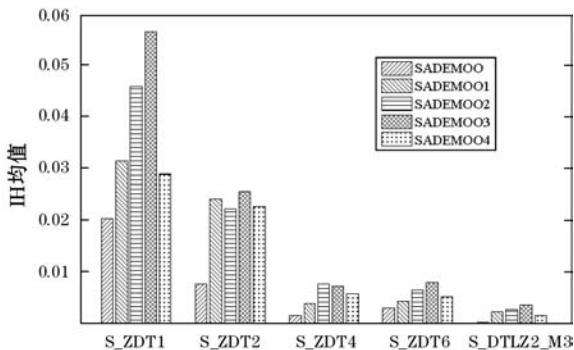
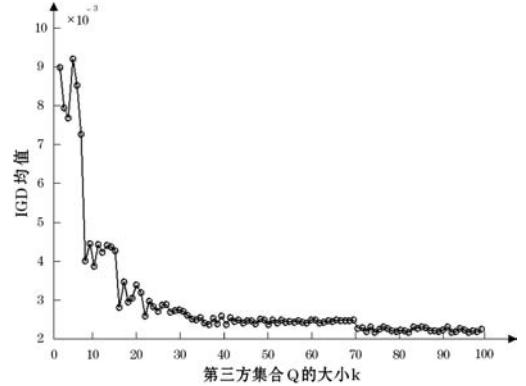


图 2 不同缩放因子调整策略在 IH 指标上对算法性能影响

结果显示,自适应选择缩放因子分布模型的策略得到的 IGD 和 IH 均值均最小,单一的高斯分布不能兼顾约近 Pareto 解集对完整性和均匀性的需求,而均匀分布的方法虽然能生

成宽广范围的缩放因子取值,但是其不能根据当前种群中个体的具体需求而实时调整,从而也会影响算法的性能。

本文还测试了第三方集合 Q 的大小 k 对算法性能的影响,此处以 S_ZDT1 为测试对象,以 IGD 指标均值为评价依据,将 k 值由 1 逐渐增加到 100(即等于存档 A 的大小)。第三方集合 Q 的大小 k 对算法性能的影响如图 3 所示。

图 3 第三方集合 Q 的大小 k 对算法性能的影响

从图 3 的测试结果显示,随着集合大小的增加,IGD 指标的取值逐渐减小,但是 k 取值越大,IGD 指标下降的幅度却逐渐减小。当 k 取值 100 时,IGD 指标取值最小,但是其计算耗时却显著增加。相比于 k 取值 20 时的情况,IGD 指标均值下降为原先的 70.97%,但在同一台计算机上的运行时间却增大到原先的 23.82 倍。这与本文在保证一定优化性能下尽量减少耗时的初衷相违背,通过测试, k 在 10 到 40 之间取值较为合理。

4 结语

为提高差分进化算法在多目标优化中性能,提出一种新的自适应多目标差分进化算法,在 5 个复杂和具有代表性的多目标优化问题上取得了好于已有自适应算法的测试结果。此外,实验结果还表明,提出的缩放因子分布模型自适应选择的策略比单一高斯或均匀分布模型更为有效,它可以兼顾约近 Pareto 解集的完整性和均匀性要求,有力地平衡了局部搜索和全局搜索的强度;而提出的基于第三方集合和优胜累积量的个体比较策略可以在保证求解质量的前提下,大幅减小算法中频繁比较所引起的过高时间代价。还通过实验的方式给出了第三方解集的大小对算法性能的影响规律,使该参数易于选取。

参考文献

- [1] 毕晓君,刘国安,肖婧.基于新变异策略的动态自适应差分进化算法[J].计算机研究与发展,2012,49(6):1288-1297.
- [2] Das S, Suganthan P N. Differential Evolution: A Survey of the State-of-the-art [J]. IEEE Transactions on Evolutionary Computation, 2011, 15(1): 4-31.
- [3] 卢金鑫.基于全局和局部搜索的整型权值神经网络混合学习算法[J].计算机应用与软件,2014,31(2):287-292.
- [4] 刘波,王凌,金以慧.差分进化算法研究进展[J].控制与决策,2007,22(7):721-729.
- [5] 熊伟丽,陈敏芳,张乾,等.基于改进差分进化算法的非线性系统模型参数辨识[J].计算机应用研究,2014,31(1):124-127.

对于 d1291 实例,算法最终都收敛于同一个局部最优解 53 101,误差为 4.53%,其周游路线如图 1 所示。对于 F11400 实例,算法最终都收敛于同一个局部最优解 20 392,误差仅为 1.32%,该似解优于文献[2]中的结果,其周游路线如图 2 所示。

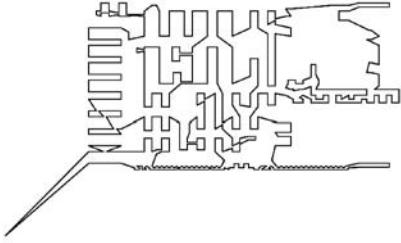
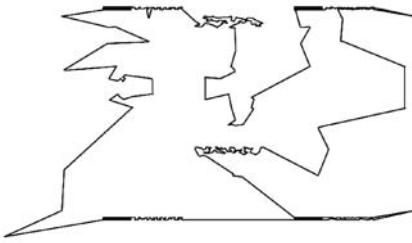


图 1 d1291 问题 SAHPA 找到的周游距离为 53 101 的周游路线



2 F11400 问题 SAHPA 找到的周游距离为 20 392 的周游路线

虽然实例 F11400 比 d1291 的规模大很多,但从图 1 和图 2 的节点分布看实例 F11400 的城市分布基本属于单环形分布,而实例 d1291 属于多环形分布,故局部最优解对算法影响较大,这就是为什么规模小的反而误差较大的原因。从图 1 和图 2 可以看出,虽然算法获得的近似解不是全局最优解,但是其周游路线依然比较完美,没有交叉点。说明对于这两个求解实例,算法在较短时间内,获得了与已知最优解误差较小的高质量近似解。

上述分析都说明了自适应混合粒子群优化算法求解较大规模 TSP 问题的可行性和有效性。

6 结 语

本文针对改进增强型自探索粒子群优化算法求解大规模 TSP 问题的不足,提出了自适应混合粒子群优化算法求解较大规模 TSP 问题,通过多种自适应策略,分阶段自适应进化,算法获得解的质量明显提高。通过对多个标准 TSP 问题的仿真测试,实验结果对比分析说明了自适应混合粒子群优化算法求解大规模旅行商问题能够的获得高质量的解。但本文算法仅仅利用了多个局部最优解的交集信息,如何更进一步利用多个局部最优解隐藏的信息,并形成有效的启发式规则以高效求解更大规模的 TSP 问题是今后要研究的重点。

参 考 文 献

[1] 冯翔,马美怡,虞慧群. TSP 湖水能量优化算法[J]. 计算机研究与发展,2013,50(9):2015-2027.
 [2] 冀俊忠,黄振,刘春年,等. 基于多粒度的旅行商问题描述及其蚁群优化算法[J]. 计算机研究与发展,2010,47(3):434-444.
 [3] 苏晓勤,孙鹤旭,潘旭华. 改进蜂群算法的旅行商问题仿真[J]. 计算机工程与设计,2013,34(4):1420-1424.
 [4] 安晶,徐森. 一种结合粒子群优化理论改进的郭涛算法及其应用[J]. 计算机应用与软件,2014,31(2):296-299,320.

[5] 李熠,马良. 用量子蚁群算法求解大规模旅行商问题[J]. 上海理工大学学报,2012,34(4):355-358.
 [6] 贾瑞玉,李亚龙,菅玉勇. 求解旅行商问题的混合量子蚁群算法[J]. 计算机工程与应用,2013,49(22):36-39.
 [7] 饶卫振,金淳,黄英艺. 求解 TSP 问题的最近邻与插入混合算法[J]. 系统工程理论与时间,2011,31(8):1419-1428.
 [8] 彭斌,胡常安,邵兵,等. 求解 TSP 问题的混合杂草优化算法[J]. 振动测试与诊断,2013,33(S1):52-55.
 [9] 王翠茹,张江维,王玥,等. 基于改进粒子群优化算法求解旅行商问题[J]. 华北电力大学学报,2005,32(6):47-51.
 [10] 熊伟,张江维,张火林. 增强型自探索粒子群优化算法求解 TSP 问题[J]. 华北电力大学学报,2009,36(6):69-74.
 [11] Zhang Jiangwei, Si Wenjian. An Improved Enhanced Self-Tentative Particle Swarm Optimization Algorithm for TSP [C]//Sixth international conference on natural computation, Yantai, Shandong, China, 2010, Piscataway, IEEE computer society, 2010:279-283.
 [12] 刘艳娟,谢晓钢,陈胜达. 一种改进的求解 TSP 问题的近似算法[J]. 计算机工程与应用,2006,42(33):71-73.

(上接第 252 页)

[6] 陈秋红,景伟娜,吴长泽. 满足可靠性约束的冗余服务池多目标优化策略[J]. 计算机应用研究,2012,29(10):3791-3793.
 [7] 黄敏,江渝,毛安,等. 基于全局最优位置自适应选取与局部搜索的多目标粒子群优化算法[J]. 计算机应用,2014,34(4):1074-1079.
 [8] Brest J, Greiner S, Boskovic B, et al. Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems [J]. IEEE Transactions on Evolutionary Computation, 2006,10(6):646-657.
 [9] Qin A K, Huang V L, Suganthan P N. Differential evolution algorithm with strategy adaptation for global numerical optimization [J]. IEEE Transactions on Evolutionary Computation, 2009,13(2):398-417.
 [10] Huang V L, Qin A K, Suganthan P N. Multi-objective optimization based on self-adaptive differential evolution algorithm [C]//The 2007 IEEE Congress on Evolutionary Computation (CEC 2007). Piscataway, NJ: IEEE Press, 2007:3601-3608.
 [11] 邱威,张建华,刘念. 自适应多目标差分进化算法在计及电压稳定性的无功优化中的应用[J]. 电网技术,2011,35(8):81-87.
 [12] 宋通,庄毅. 基于多种群差分进化的多目标优化算法[J]. 计算机科学,2012,39(8):205-209.
 [13] Corne D W, Jerram N R, Knowles J D, et al. PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization [C]//The Genetic and Evolutionary Computation Conference (GECCO'2001). USA: Morgan Kaufmann, 2001:283-290.
 [14] Huang V L, Qin A K, Deb K, et al. Problem definitions for performance assessment of multi-objective optimization algorithms [R]. Special Session on Performance Assessment of Multi-Objective Optimization Algorithms, Nanyang Technological University, Singapore, 2007.
 [15] Zhou Ai-min, Zhang Qing-fu, Jin Yao-chu. Approximating the set of Pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm [J]. IEEE Transactions on Evolutionary Computation, 2009,13(5):1167-1189.
 [16] Zitzler E, Thiele L, Laumanns M, et al. Performance assessment of multiobjective optimizers: An analysis and review [J]. IEEE Transactions on Evolutionary Computation, 2003,7(2):117-132.