

文章编号:1001-9081(2010)11-2887-04

自适应多目标混合差分进化算法在联盟运输调度中的应用

蔡延光¹, 宋康¹, 张敏捷¹, 武鑫²

(1. 广东工业大学 自动化学院, 广州 510006; 2. 红河钢铁有限公司, 云南 红河 661100)

(songk518@163.com)

摘要:传统的单目标算法运行一次只能得到一个解,而多目标算法运行一次可以得到一个解集。文中所提算法(DEASA)通过改进差分进化策略,设计重构,调整自适应参数,并采用擂台法则构建非支配集,将模拟退火策略融入差分进化算法当中,进一步提高了算法的性能,降低了时间复杂度,增强避免陷入局部最优的能力。通过实验验证表明,该算法能有效地解决联盟运输调度问题。

关键词:联盟运输调度;差分算法;模拟退火;非支配集;多目标最优化

中图分类号: TP18 **文献标志码:** A

Adaptive multi-objective hybrid differential evolution algorithm in union transport scheduling

CAI Yan-guang¹, SONG Kang¹, ZHANG Min-jie¹, WU Xin²

(1. Automation College, Guangdong Technical University, Guangzhou Guangdong 510006, China;

2. Honghe Iron and Steel Shares Company Limited, Honghe Yunnan 661100, China)

Abstract: The traditional single-objective algorithm can only get one solution, but the multi-objective algorithm can get a solution set after every run. The algorithm (DEASA) improved the differential evolution strategy, designed reconstruction, adjusted parameter adaptively, adopted arena's principle to build non-dominated set rules, and added simulated annealing strategies into the differential evolution algorithm which enhanced the ability to further improve the performance of the algorithm and reduced the time complexity to avoid falling into local optimum. The experiments show that this algorithm can effectively solve the union transport problem.

Key words: union transport scheduling; difference algorithm; simulated annealing; non-dominated set; multi-objective optimization

0 引言

联盟运输调度问题即基于物流联盟的运输调度最优化,就是在满足运输要求的前提下(如运输车辆限制、运输时间限制、运输数量要求等),在联盟内快速组织多种交通工具及适当的行车路线和任务分配,排出每辆车的作业流程和各个任务的送货时间、数量一览表,允许车辆中转,达到一定的目标(如费用极小、路程最短、时间最少、使用车辆数尽量少等),追求经济效益的最大化和实现过程最优化^[1]。

目前,出现了很多种不同的多目标进化算法,比如 PESA、PAES、SPEA、SPEA2、NSGA2、MOMGA、MOMGA-II等。其绝大多数是通过构造非支配集,然后对非支配集中的个体进行选择,交叉,变异等操作,从而得到下一代的种群。在以上这些多目标进化算法中,现已被证明收敛速度最快的是 NSGA2。但是其构造非支配集的时间复杂度为 $O(rN^2)$ 。众所周知,确定一个非支配集的时间复杂度的下界是 $O(rN \log N)$,所以 NSGA2 并不是最优的。在本文采用文献[2]提出的擂台法则,这种构造非支配集的方法结合了擂台竞赛规则的思想,在最坏的情况下能将时间复杂度降低到 $O(rmN)$ 。在一般情况下 m 总是比 $N/2$ 小,所以擂台法则具有较好的实用价值。

差分进化算法(Differential Evolution, DE)是 Dainer Storn 和 Kenneth Price 于 1995 年共同提出的一种采用浮点矢量编码,在连续空间中进行启发式随机搜索的优化算法^[3]。差分进化算法因原理简单、受控参数少、鲁棒性强等特点^[4],引起越来越多的学者关注。

1 数学模型的建立

带时间窗约束的车辆路径问题(Vehicle Routing Problem with Time Windows, VRPTW)描述如下:网络 $G(V, A)$ 中(V 代表车辆的集合, A 代表路径节点的集合)共有 N 个客户,1 个中心仓库(记做客户 0)。已知 $A = \{i | i = 0, 1, 2, \dots, N\}$ 为 N 个客户点的集合, $i = 0$ 时为中心仓库; $V = \{k | k = 1, 2, \dots, K\}$ 表示车辆集合; i, j 为顾客数($i = 1, 2, \dots, N; j = 1, 2, \dots, N$); $x_{ijk} = 1$ 为车辆 k 服务客户 i 后直接行驶到顾客 j 为其服务, $x_{ijk} = 0$ 为其他; d_{ij} 为从客户 i 到客户 j 的距离; e_i 为到达客户 i 处规定的最早到达时间; l_i 为到达客户 i 处规定的最晚到达时间; t_{ik} 为第 k 辆车到达客户 i 处的时间; m_i 为客户 i 处的产品需求量; Q 为每辆车的载重最大量; t_{ij} 为从客户 i 到客户 j 处的运输时间; s_i 为车辆在客户 i 处的服务时间; T 是个很大的数。

目标为:

收稿日期:2010-04-28;修回日期:2010-07-28。

基金项目:广东省自然科学基金资助项目(8351009001000002);广东省科学技术厅项目(2007B010200070)。

作者简介:蔡延光(1963-),男,湖北咸宁人,教授,博士,主要研究方向:组合优化、人工智能、决策支持系统;宋康(1984-),男,湖南衡阳人,硕士,主要研究方向:物流优化与控制;张敏捷(1984-),男,湖北咸宁人,硕士,主要研究方向:智能交通;武鑫(1985-),男,云南永善人,主要研究方向:智能控制。

$$\min Z_1 = \sum_{k=0}^K \sum_{i=0}^N \sum_{j=0}^N d_{ij} x_{ijk} \quad (1)$$

$$\min Z_2 = \sum_j \sum_k x_{0jk} \quad (2)$$

约束条件为:

$$\sum_{k=1}^K \sum_{j=0, j \neq i}^N x_{ijk} = 1; i = 1, 2, \dots, N \quad (3)$$

$$\sum_{i=1}^N m_i \sum_{j=0}^N x_{ijk} \leq Q; \forall k \in K \quad (4)$$

$$\sum_{j=0}^N x_{0jk} = 1; \forall k \in K \quad (5)$$

$$\sum_{i=0}^N x_{ihk} - \sum_{j=0}^N x_{hjk} = 0; \forall k \in K, \forall h \in N \quad (6)$$

$$\sum_{i=0}^N x_{aik} = 1; \forall k \in K \quad (7)$$

$$t_{ik} + s_i + t_{ij} + \max[(e_i - t_{ik}), 0] - T(1 - x_{ijk}) \leq t_{jk} \quad (8)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ijk} \leq |S| - 1; \forall k \in V, S \subseteq A \quad (9)$$

$$x_{ijk} = \begin{cases} 1, & \text{车辆 } k \text{ 从点 } i \text{ 到点 } j; i, j \in A; k \in V; i \neq j \\ 0, & \text{其他} \end{cases} \quad (10)$$

式(1)所示的目标函数表示车辆运行的总路径最少;式(2)所示的目标函数表示所需车辆数最少;式(3)表示每个客户只能由一辆车来服务,且每个客户都能服务而且只能服务一次;式(4)表示每辆车所运输的货物重量都不能超过车辆重量的最大载重量;式(5)~(7)表示每辆车从中心仓库出发,经过若干个不重复的客户点后,最后又回到中心仓库点;式(8)表示第 k 辆车到达客户 i 的时间加上在客户 i 处服务的时间,加上从客户 i 到客户 j 行驶的时间,再加上在客户 j 处等待的时间,小于或等于第 k 辆车到达客户 j 的时间;式(9)为支路消去约束,保证运输路线不形成内部闭环;式(10)为决策变量。

2 算法的设计

2.1 多目标优化的相关概念

通常在多目标优化领域中广泛采用并普遍接受的多目标优化问题(Multi-objective Optimization Problem, MOP)的数学定义如下^[5]。

定义1 MOP。一般MOP由 n 个决策变量参数、 k 个目标函数和 m 个约束条件组成,目标函数、约束条件与决策变量之间是函数关系。最优化目标如下:

$$\begin{aligned} \text{Maximize } y &= f(x) = (f_1(x), f_2(x), \dots, f_k(x)) \\ \text{s. t. } e(x) &= (e_1(x), e_2(x), \dots, e_m(x)) \leq 0 \end{aligned} \quad (11)$$

其中:

$$x = (x_1, x_2, \dots, x_n) \in X$$

$$y = (y_1, y_2, \dots, y_n) \in Y$$

x 表示决策向量, y 表示目标向量, X 表示决策向量 x 形成的决策空间, Y 表示目标向量 y 形成的目标空间,约束条件 $e(x) \leq 0$ 确定决策向量的可取行取值范围。

定义2 Pareto最优解。对于集合 $A \subseteq X_f$,决策向量 $x \in X_f$ 为非劣的当且仅当:

$$\nexists a \in A: a > x \quad (12)$$

即当且仅当 x 在 X_f 是非劣的,决策向量 x 才是Pareto最优解。

2.2 多目标自适应混合差分进化算法(DEASA)求解

2.2.1 种群的初始化及适应度函数

为保证群体的多样性,在种群的初始化阶段,采用随机生

成个体,按照前向插入法^[6](Push-Forward Insertion Heuristic, PFIH)的思想生产个体,解码时只需按序不断地将基因位的值表示的节点插入到路线中去即可。当某一个点不满足约束时(载重约束、时间窗或其他条件),就新增一条路径并把这个点插入,即按照路线数越少,访问的总费用可能越少的思想,尽可能地向原有的路线中插入客户点。例如染色体串8 2 5 3 7 1 9 6 4,经过路径译码为:

路线1:0→8→2→5→3→0。

路线2:0→7→1→9→0。

路线3:0→6→4→0。

针对两个目标,适应度函数分布为:

$$\min Z_1 = \sum_{k=0}^K \sum_{i=0}^N \sum_{j=0}^N d_{ij} x_{ijk}$$

$$\min Z_2 = \sum_j \sum_k x_{0jk}$$

这比随机生成个体的多样性相对提高了。按照PFIH策略生成的个体占很少一部分,剩下的大部分个体由随机方式生成。

2.2.2 变异选择操作

DE/rand/1/bin和DE/best/1/bin是目前使用最广、应用最为成功的差分策略。前一种策略有利于保持群体的多样性,后一种策略则更强调算法的收敛速度。

为了并行执行两种差分策略,先将父代种群按照一定的比例分成两个子种群,两个子种群分别通过指定的差分策略产生变异个体,从而实现差分策略的混合。本文采用随机分配方法得到两个子种群。假设DR为预先指定的分配系数。子种群分配及差分策略混合的过程如下:

For $i = 1 : \text{NP}$

$j = j_{\text{rand}} \neq i$

If $\text{rand}(0, 1) < \text{DR}$

$x_i(g)$ 分配到子种群1, $x_j(g)$ 作为 $x_i(g)$ 的扰动中心,

按 DE/rand/1 产生变异个体

Else

$x_i(g)$ 分配到子种群2, $x_b(g)$ 作为 $x_i(g)$ 的扰动中心,

按 DE/best/1 产生变异个体

End

End

DE采用“贪婪”的搜索策略,经过变异和交叉操作后生成的实验个体 x_i 和 x_j^g 进行竞争。只用当 x_i 的适应度比 x_j^g 更优时才被选作子代;否则直接将 x_j^g 作为子代,选择操作的公式为:

$$x_j^{g+1} = \begin{cases} x_i, & f(x_i) < f(x_j^g) \\ x_j^g, & f(x_i) \geq f(x_j^g) \end{cases} \quad (13)$$

2.2.3 聚集度分析及种群重构

设 $f(x_b(g))$ 表示第 g 代种群中最佳个体的适应度; $\overline{f(g)}$ 表示第 g 代种群的平均适应度,则群体聚集度因子可定义为^[7]:

$$c = \min(f(x_b(g)), \overline{f(g)}) / \max(f(x_b(g)), \overline{f(g)}) \quad (14)$$

其中: $c \in (0, 1)$,它反映出了种群当前的聚集程度,同时也在一定程度上也反映了种群中个体的多样性。 c 值越大,聚集程度越高,个体差异性越小。当聚集度因子小于某个阈值 R 时,执行重构操作重新初始化种群。

种群重构是将发生早熟收敛的种群通过外部作用在一定

有效范围内重新分布个体的位置,增加种群的多样性,扩展算法的搜索范围。

Blackwell 和 Branke 根据量子力学中电子的量子状态^[8],在最优个体附近随机产生重构个体,重构后的种群虽然分布于最优个体的周围,但搜索范围大到整个空间。为了能充分利用种群重构前的搜索信息,缩小种群重构后的搜索范围,加快算法的收敛速度,提出种群重构范围自适应缩小方案,即:

$$x_{i,j}(g+1) = x_{i,j}(g) + s^g \text{rand}(0,1)(x_i^u - x_j^l) \quad (15)$$

其中:收缩系数 $s \in (0,1)$, g 为进化代数。显然,重构后的种群范围与进化代数有关,随着进化代数的增加,重构后的种群范围越来越小,有利于提高局部搜索效率。

2.2.4 模拟退火策略

模拟退火算法^[9]是由 Metropolis 等人 1953 年提出的基于 MonteCarlo 迭代求解策略的一种随机寻优算法,它的思想来源是固体物质退火的过程,该算法设置一个初始温度和一个退火控制参数。模拟退火由某一较高初始温度开始,利用具有概率突跳特性的 Metropolis 抽样策略在解空间中进行随机搜索,伴随温度的不断下降重复抽样过程,最终得到问题的全局最优解。模拟退火算法在初始温度足够高、温度下降足够慢的条件下,能以概率 1 收敛到全局最优值。由于它以某种概率接受较差点,从而具有跳出局部最优解的能力;并且理论上已经证明了模拟退火算法是全局最优算法,但前提条件是选择适当的退火控制参数。

模拟退火算法的基本操作步骤如下(以求函数的最小值为例)。

第 1 步 给定初始温度 T 、退火控制参数 k 及初始点 x , 计算该点的函数值 $f(x)$ 。

第 2 步 随机产生扰动 Δx , 得新点 $x^* = x + \Delta x$, 计算新点的函数值 $f(x^*)$, 则函数值的差:

$$\Delta f = f(x^*) - f(x)$$

第 3 步 Metropolis 准则:若 $\Delta f \leq 0$, 则接受新点,并把该点作为下一次迭代的初始点;若 $\Delta f > 0$, 计算新点接受概率为 $P(\Delta f) = \exp(-\Delta f/T)$, 再产生一个区间 $[0,1]$ 上的伪随机数 r , 若 $P(\Delta f) \geq r$, 则接受新点作为下一次迭代的初始点;否则,放弃新点,仍取原来的点作为下一次迭代的初始点,执行降温操作 $T = T \cdot k$, 返回第 2 步。

2.2.5 自适应策略

当种群内的个体适应度趋于一致或者收敛于局部最优解时,使 F 增加;而当群体适应度比较分散时,使 F 减小。同时对于适应度大于平均适应度的个体,对应大的 F 值,使该解被淘汰掉;相反对于适应度低于平均适应度的个体,越接近平均适应度的个体对应越大的 F 值,从而确保多样性。因此,自适应的缩放因子能够提供相对于每个解的最佳 F 值。自适应差分进化算法在保持群体多样性的同时,保证差分进化算法的收敛性。按式(16)进行自适应调整:

$$F = \begin{cases} 1 - \frac{f_{\text{avg}} - f'}{f_{\text{avg}} - f_{\text{best}}}, & f' < f_{\text{avg}} \\ 1, & f' \geq f_{\text{avg}} \end{cases} \quad (16)$$

其中: f' 为待变异个体的适应度值; f_{avg} 为群体的平均适应度值; f_{best} 为群体中最大的适应度值。算法开始阶段 f_{avg} 和 f_{best} 差距很大,所以几乎不存在局部收敛的可能,当 f' 越小,即 F 越小,良好的基因可能会保持下来。随着演化代数的增加 f_{avg} 和

f_{best} 差距减小, F 有减小的趋势,向最优解收敛的速度逐步加快。由于收敛速度是逐步加快的,所以减少了局部收敛的危险。

在基本差分进化算法中。交叉算子 CR 通常是事先取定的一个 0 到 1 之间的实数。交叉过程如下:

$$u_{j,i}(g+1)' = \begin{cases} u_{j,i}(g+1), & \text{rand}(0,1) \leq CR \text{ 或 } j = j_{\text{rand}} \\ x_{j,i}(g), & \text{其他} \end{cases} \quad (17)$$

其中 CR 为交叉概率。 CR 用来控制 $u_{j,i}(g+1)$ 和 $x_{j,i}(g)$ 对 $u_{j,i}(g+1)'$ 的贡献,其大小在计算过程中保持不变。但是在实际搜索进程当中,若 CR 值较大,则 $u_{j,i}(g+1)$ 对 $u_{j,i}(g+1)'$ 贡献较多,此时算法的局部搜索能力较强,收敛速度较快;若 CR 值较小,则 $x_{j,i}(g)$ 对 $u_{j,i}(g+1)'$ 贡献较多,此时种群的多样性较好,算法全局搜索能力较强。良好的 CR 应在搜索的初始阶段较小以保持种群的多样性,此时算法进行全局搜索,可有效避免早熟现象发生;而在搜索的后期其值应随着搜索进程逐步增大,提高算法的局部搜索能力。基于这种思想,改进交叉概率因子,它由 3 段构成: CR 先保持最小;然后由小变大过渡;最后保持接近 1 的最大。这样,协调好何时进行全局搜索,何时进行局部搜索,以及搜索时的强度,使全局搜索能力与局部搜索能力良好平衡,快速收敛得到最优解。采用文献[10]提出的改进的交叉概率因子:

$$CR = CR_{\min} + (CR_{\max} - CR_{\min}) \times \exp\left(-30\left(1 - \frac{g}{G}\right)^t\right) \quad (18)$$

其中: t 为大于 1 的整数(本文取值为 3), g 为当前迭代次数, G 为最大迭代次数,参数 $CR_{\min} = 0.25$, $CR_{\max} = 0.85$ 。随着进化代数的增加, CR 线性减小。这种参数自调整的目的,是希望 DE 算法在搜索初期能够保证较大的种群多样性,而到后期具有较快收敛速度。

2.2.6 构造非支配集

用擂台法则^[2]构造一个进化群体的 Pareto 最优解集时,每次搜索新的非支配集个体时不需要与已有的非支配个体进行比较,每一轮比较时在构造集中选出一个个体出任擂主(一般为当前构造集的第一个个体),由擂主与构造集中的其他个体进行比较,败者被淘汰出局,胜者成为新的擂台主,并继续该轮比较;一轮比较后,最后的擂台主个体即为非支配个体。按照此方法进行下一轮比较,直至构造集为空。

擂台法则的基本思想是:设 Pop 为一进化种群, Q 为构造集,初始时 $Q = Pop$; Nds 为非支配集,初始时为空。从 Q 中任取一个个体 x 作为擂主,依次与 Q 中所有其他个体 y 比较。如果 x 支配 y ,则将个体 y 从 Q 中清除;如果 y 支配 x ,则用 y 代替 x 成为擂主,并继续与 Q 中的个体进行比较。一轮比较后,将擂主并入非支配集中。依此下去,直至 Q 为空。

2.2.7 算法框架

本文提出的 DEASA 在 DE 算法的基础上根据求解进程自适应的调整缩放因子 F 、交叉率 CR ,有效避免算法早熟收敛以及加快算法的收敛速度。

DEASA 的具体算法步骤如下:

第 1 步 确定种群大小 N ,初始化各种参数。

第 2 步 按照 2.2.1 节初始化种群及适应度评价。

第 3 步 按照式(16)和式(18)计算当前的 F 和 CR 。

- 第 4 步 执行差分进化算法的交叉、变异、选择操作,得到实验个体。
- 第 5 步 判断聚集度 c , 是否进行重构。
- 第 6 步 模拟退火操作,应用 Metropolis 准则决定是否接受实验个体。
- 第 7 步 降温操作, $T = T \cdot k$, k 是一个在 0 和 1 之间的常数。
- 第 8 步 用擂台法非支配集构造。
- 第 9 步 算法终止条件判断,如果满足终止条件,则输出结果,算法结束;否则转第 2 步。

3 实验分析与计算

为了验证算法的可行性和有效性,对一个有 16 个客户 AVRP 进行求解,物流中心和客户均分布在 100×100 的范围内,各数据见表 1,配送车辆的最大载重量为 30,要求组织适当的行车路线,使总配送路径距离最短及车辆数最少。

表 1 客户数据

客户	位置	需求量 q_i	客户	位置	需求量 q_i
1	(72, 30)	5	9	(75, 83)	10
2	(61, 36)	4	10	(26, 36)	5
3	(62, 70)	9	11	(53, 45)	3
4	(89, 12)	6	12	(33, 15)	3
5	(31, 15)	8	13	(35, 82)	14
6	(62, 22)	9	14	(31, 64)	15
7	(82, 82)	8	15	(65, 17)	5
8	(56, 28)	5	16	(61, 38)	5

实验用机 CPU 是酷睿双核 2.0 GHz。所有算法用 VC++ 语言编写,运行于 Windows XP 系统。参数设置:种群规模 $N = 100$, 预定分配系数 $DR = 0.7$, 参数 $CR_{\min} = 0.25$, $CR_{\max} = 0.85$, $R = 0.4$, $s = 0.9$, $t = 3$, G 最大迭代次数为 1 000, 初始温度 $T = 260\text{ }^{\circ}\text{C}$ 及降温控制参数 $k = 0.94$ 。

DEASA 经过计算,求得最好解如下。
物流中心总共需 4 辆车,各配送路径如下:

- 配送路径 1: 0→13→14→0。
- 配送路径 2: 0→3→9→7→0。
- 配送路径 3: 0→10→5→12→8→16→11→0。
- 配送路径 4: 0→6→15→4→1→2→0。

总路径长度为: 417.65 km, 具体如表 2 所示。

最优配送方案路径图如图 1 所示(黑圆点表示客户)。

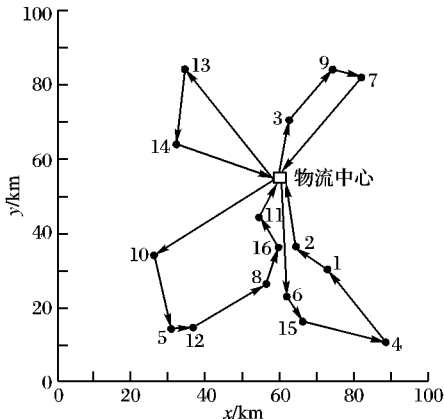


图 1 最优配送方案的配送线路

对于此问题算例,还采用了文献[11]提出的构造非支配

方法 NSGA2 进行了仿真。表 3 是本文提出的构造非支配集方法和文献[11]提出构造非支配集的方法的运行效率比较。由此表可知,本文算法的运算效率明显提高了。

表 2 配送路径及其里程

线路	车辆路径	实载量/车辆容量	总里程
1	0→13→14→0	29/30	93.97
2	0→3→9→7→0	27/30	72.66
3	0→10→5→12→8→16→11→0	29/30	130.82
4	0→6→15→4→1→2→0	29/30	120.20
合计		114/120	417.65

表 3 算法运行时间的比较

种群大小 N	NSGA2	DEASA
200	5.22	4.27
400	23.550	15.330
600	43.110	27.912
800	73.280	42.420
1 000	119.330	72.360

4 结语

传统的单目标算法只能解决单一的问题,但现实生活中更多的是需同时处理多个目标。本文将联盟运输调度问题描述为一个多目标问题,构建以配送路径最短、所需车辆数最少为目标的新模型。本文算法通过改进差分进化策略,设计重组、自适应参数调整,并采用擂台法则构建非支配集,将模拟退火策略融入到差分进化算法当中,很好地解决了此模型的求解。多目标混合差分算法的多个解可以为决策者提供更多的选择余地,作出更好的决策。该模型的建立有助于有效地解决 VRPTW,提高了物流工作效率,为城市交通质量提高提供了有力的保障。

参考文献:

[1] 邹谷山,蔡延光,王涛. 运动调度问题的现状与发展方向[J]. 城市交通,2005,3(1): 47-50.

[2] 邝达,郑金华. 一种快速构造非支配集的方法——擂台法则[J]. 高技术通讯,2005,15(4): 11-14.

[3] STORE R, PRICE K. Differential evolution — A simple and efficient adaptive scheme for global optimization over continuous spaces [EB/OL]. [2009-12-12]. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.67.5398&rep=rep1&type=pdf>.

[4] 杨启文,蔡亮,薛云灿. 差分进化算法综述[J]. 模式识别与人工智能,2008,28(4): 506-510.

[5] 崔逊学. 多目标进化算法及其应用[M]. 北京: 国防工业出版社,2006.

[6] 刘敏. 多目标遗传算法在车辆路径优化中的应用研究[D]. 湘潭: 湘潭大学,2006.

[7] 张选平,杜玉平,秦国强,等. 一种动态改变惯性权的自适应粒子群算法[J]. 西安交通大学学报,2005,39(10): 1039-1042.

[8] TIM B, JURGEN B. Multi-swarm optimization in dynamic environments [EB/OL]. [2009-12-12]. <http://www.springerlink.com/content/7mm505r4rc16qab/fulltext.pdf>.

[9] 都志辉,李三立,吴梦月. 混合 SPMD 模拟退火算法及其应用[J]. 计算机学报,2001,24(1): 91-98.

[10] 邓泽喜,曹敦虔,刘晓冀,等. 一种新的差分进化算法[J]. 计算机工程与应用,2008,44(24): 40-42.

[11] DEB K, PRATAP A, AGRAWAL S, et al. A fast and elitist multi-objective genetic algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182.