

文章编号:1000-6788(2006)11-0130-09

基于实数编码遗传算法的空车流量分配优化方法研究

张喜,张全寿

(北京交通大学 交通运输学院系统工程与控制研究所,北京 100044)

摘要: 针对铁路空车流量分配优化问题求解的复杂性,提出了采用实数编码遗传算法(Real-coded Genetic Algorithm,简称:RGA)求解的智能优化方法.对空车流量分配优化问题的RGA算法的基本结构、个体与适应度函数的构造、遗传算子的设计和算法的基本特点等进行了深入研究,并结合铁路空车流量分配问题的实例分析验证了算法的有效性.

关键词: 空车流量分配;实数编码;遗传算法

中图分类号: U292.6

文献标志码: A

Study on the Real-coded Genetic Algorithm of the Empty Wagons Flow Assignment

ZHANG Xi, ZHANG Quan-shou

(School of Traffic and Transport, Beijing Jiaotong University, Beijing 100044, China)

Abstract: Taking account of the complexity of the solution of empty wagon flow assignment optimization, the paper puts forward an intelligent optimized method—Real-coded Genetic Algorithm(RGA). It deals with the basic structure of RGA and formation of individual and fitness function, analyzes the basic characteristics of the algorithm and designs the genetic seed. Finally, this paper uses an example of empty wagon flow assignment to test the effectiveness of the algorithm.

Key words: empty wagons flow assignment; Real-coded; genetic algorithms

0 引言

在铁路货物运输网络系统中,最理想的空车空间分布应是保证任何时间段到达各车站(或区域)卸车产生的空车数,正好是各车站(或区域)装车所需要的装车数,即保证任何时间段各车站(或区域)装卸车辆数均衡.它不但可保证最好地满足运输需求和有节奏地完成运输任务,而且能最经济有效地发挥车辆的运输效率.然而,由于受货物运送的单向性、区域间运输需求量的不平衡性,以及季节性波动或其他干扰的影响,实际上这种理想的均衡是很难保证的.为了消灭或减少各车站(或区域)装卸车辆数不均衡情况,在交通规划、车流组织和运输计划中必须考虑空车流量分配优化问题.

对空车流量分配优化问题的研究,通常都可归结为一个带约束的单目标非线性数学规划模型.目前对该模型的求解仍存在两大困难:其一是计算上的困难,由于模型的决策变量是路网上众多弧上的车流量,且模型要求的解具有整数性,属于复杂整数规划问题,其形式在组合优化和计算理论中被归为NP完全问题类^[1],所以一般没有可供采用的多项式算法以求解,通常采用的是一些启发式搜索算法,这些算法一般不能有效地利用问题的固有知识来缩小搜索空间,对规模大一些的路网寻求模型的最优解或准最优解时,往往会引起搜索的组合爆炸,而使求解计算发生困难.其二是算法构造上的困难,由于实际的空车流量分配优化模型一般是多极值非线性函数,并且不具有连续可微性,而常规的非线性优化技术,通常是基于梯度的寻优方法,要求优化问题可微,初始解必为可行解,一般只能求得局部最优解,不具有识别全局最优解的能力,所以很难采用常规非线性优化技术构造出简单有效的实用算法.

收稿日期:2005-07-11

作者简介:张喜(1953-),汉族,男,河北大名人,北京交通大学教授,工学博士.

遗传算法 (Genetic Algorithms, 简称 GA) 是一种智能优化算法. 该算法是基于“适者生存”的一种新兴的自适应随机搜索方法, 具有通用、简单、有效、鲁棒性强等特点, 而且对优化对象既不要求搜索空间具有连续性, 也不要求具备可微、单调等性能, 比较适合于对非凸空间中复杂的多极值优化问题和组合优化问题的求解^[2]. 应用遗传算法求解复杂的带约束的非线性数学规划问题, 具有既可减少问题求解计算上的难度, 又可提高解的全局最优性和求解计算的效率等特点. 本文针对空车调整流量分配优化模型的求解问题, 对遗传算法的应用进行了深入研究.

1 空车流量分配优化问题数学模型的构造

定义 空车流量分配运输网络(路网)为图 $G\{V, E, B_{(e)}, C_{(e)}\}$, 其中: $V = \{1, 2, \dots, N\}$, 表示路网中节点的集合; $E = \{e = \{e_{\text{正}} = (i, j), e_{\text{反}} = (j, i)\} \mid i, j \in V, e = 1, 2, \dots, M\}$, 表示路网中相邻节点间双向路段弧(正向弧和反向弧)的集合; $B_{(e)} = \{b_e \mid e \in E, e = 1, 2, \dots, M\}$, 表示路网中路段弧上的运输能力容量集合; $C_{(e)} = \{c_e(\cdot) \geq 0, \mid e \in E, e = 1, 2, \dots, M\}$, 表示路网中路段弧上单位车流量运输总耗费集合, $c_e(\cdot) = (l_e, f_e) = l_e + (f_e)$, 表明路段弧上的耗费主要是弧长度 l_e 和弧流量 f_e 的函数.

设已知路网中 O-D 对 i, j 间的空车调配需求量为 x_{ij} ($x_{ij} \geq 0, i, j \in V$); 车流可能运行的合理路径 l_{ij} 条, 合理路径集为 $P_{ij} = \{p_{ij}^k \mid k = 1, 2, \dots, l_{ij}, i, j \in V\}$ ^[1].

铁路空车流量分配优化问题, 通常可采用路网的弧-路关系描述为如下的数学模型^[3].

目标函数:

$$\begin{aligned} \min Z_1 &= \sum_{ij} \sum_{k=1}^{l_{ij}} f_{ij}^k \sum_{e \in E} p_{ij,e}^k [l_e + (f_e)] \quad (\text{总运输耗费最少化目标}) & (1) \\ \max Z_2 &= \sum_{ij} \sum_{k=1}^{l_{ij}} f_{ij}^k \quad (\text{总分配流量最大化目标}) \\ \max Z_3 &= \sum_{ij} w_{ij} \sum_{k=1}^{l_{ij}} f_{ij}^k \quad (\text{总社会效益最大化目标}) \end{aligned}$$

约束条件:

$$\begin{aligned} g_1(f_{ij}^k) &= b_e - \sum_{ij} \sum_{k=1}^{l_{ij}} p_{ij,e}^k f_{ij}^k \leq 0, \quad \forall e \in E \quad (\text{运输能力约束}) \\ g_2(f_{ij}^k) &= x_{ij} - \sum_{k=1}^{l_{ij}} f_{ij}^k \leq 0, \quad \forall i, j \in V \quad (\text{运输供求约束}) \\ g_3(f_{ij}^k) &= f_{ij}^k \geq 0 \text{ 的整数}, \quad \forall k \in l_{ij} \quad (\text{变量非负约束}) \\ g_4(f_{ij}^k) &= f_{e_{\text{正}}} \cdot f_{e_{\text{反}}} = 0, \quad \forall e_{\text{正}}, e_{\text{反}} \in E \quad (\text{无对流运输约束}) \end{aligned}$$

式中, f_{ij}^k : O-D 对 i, j 间流经路径 p_{ij}^k 的流量变量.

$p_{ij,e}^k$: 弧-路关联变量, 若路段弧 e 在路径 p_{ij}^k 上, 则 $p_{ij,e}^k = 1$, 否则 $p_{ij,e}^k = 0$.

$f_e = \sum_{ij} \sum_{k=1}^{l_{ij}} p_{ij,e}^k f_{ij}^k, \forall e \in E$: 路段弧 e 上的总流量.

$$(f_e) = \begin{cases} B_1 f_e^2 + B_2 f_e + B_3, & 0 < f_e < b_e \\ B, & f_e > b_e \end{cases}, \quad \forall e \in E: \text{路段弧 } e \text{ 上与运距无关的单位车流量耗费函数,}$$

B_1, B_2, B_3 是曲线拟合参数, 罚参数 B 是一个充分大的正数.

w_{ij} : 发点 i 到收点 j 的空车运用分配优先权, 权值大的空车需求优先分配.

该模型综合体现了空车流量分配运输耗费(空车走行公里、运输支出费用等)最少、分配(排空)流量最大和社会效用最大的多目标优化原则. 约束条件综合反映了路网运输能力、运输需求量、决策变量取值及无同种空车对流运输要求等对空车流量分配的制约因素.

设模型的决策变量用向量形式记为 $f = (\dots, f_{ij}^k, \dots)^T$; 把三个目标函数 Z_1, Z_2, Z_3 通过线性加权和系数 $\alpha_i \geq 0 (i=1, 2, 3 \text{ 且 } \sum_{i=1}^3 \alpha_i = 1)$, 构造为单目标评价函数 $Z(f)$; 约束条件式用约束集合表示记为 R . 则空车流量分配多目标优化问题的线性加权和弧-路模型可用向量形式表示为:

$$\begin{aligned} \min Z(f) &= \alpha_1 Z_1(f) - \alpha_2 Z_2(f) - \alpha_3 Z_3(f), \\ \text{s.t. } f &\in R = \{f \in E^n \mid g_i(f) \leq 0, (i = 1, 2, 3, 4)\}. \end{aligned} \tag{2}$$

该模型的特点是决策变量数量庞大(路网上众多弧上的车流量)、变量要求取整型实数值、目标函数和约束条件中包含有决策变量的不连续非线性函数. 显然, 该优化问题是一个复杂的实数型数学规划问题, 采用常规的数学优化技术已很难有效地求解, 本文将针对该模型的求解设计构造相应的遗传算法.

2 空车流量分配优化问题实数编码遗传算法(RGA)的设计

2.1 基于实数编码的个体及适应度函数的设计

编码是设计遗传算法时要解决的首要问题, 其中编码方法的选择在很大程度上将决定算法的性能与效率. 二进制编码方法是遗传算法中最常用的一种编码方法, 或者说传统的遗传算法是基于二进制位串操作的方法. 然而, 实际工程优化问题中的决策变量往往都是实数, 如空车调整流量分配优化问题中的空车调配 O-D 量、线路输送能力及线路上分配的空车流量的取值均是整型实数. 对于这类实数变量众多且取值范围大的函数优化问题, 由于优化变量维数过高, 如采用二进制编码, 不仅会造成编码过长, 从而占用较多的计算机存储空间, 还会由于增加算法的计算量, 使得遗传算法的收敛速度降低, 甚至无法接受. 因此, 应用遗传算法求解函数优化问题时, 一般必须考虑采用实数编码技术, 建立基于实数编码的遗传算法 (Real-coded Genetic Algorithm, 简称: RGA).

关于实数编码的表示方式有如下定义^[4].

定义 1 实数编码向量个体 x_i : 由 n 个优化变量所构成的 n 维实数空间中某一点的各个变量值按规定顺序编码所组成的集合, 称之为该点对应的实数编码向量个体 (简称: 实数个体或个体), 记为 $x_i = (x_{i1}, x_{i2}, \dots, x_{in})^T, x_i \in R^n$.

定义 2 实数编码种群 (或群体) P : 由 m 个实数编码向量个体 x_i 所构成的集合, 记为 $P = \{x_1, x_2, \dots, x_m\}$. 其中, m 称为种群 P 的群体规模 (或群体大小). 在遗传算法中 P 也可用个体空间描述, 记为 $I^m = \{x_1, x_2, \dots, x_m\}$, 表示群体规模为 m 的个体空间.

定义 3 定义对应于实数编码向量个体 x_i 的适应度函数为 $f(x_i) = (Z(x_i))$. 其中, $Z(x_i)$ 为优化问题的目标函数, 不失一般性, 这里考虑 $Z(x_i)$ 为最小化问题的目标函数. $(Z(x_i))$ 为目标函数转换为适应度函数的适应变换函数, 一般不要求适应度函数 $f(x_i)$ 的值与目标函数 $Z(x_i)$ 的值相等, 但 $f(x_i)$ 必须总是以 $Z(x_i)$ 为变量的函数, 并且具有适应度函数 $f(x_i)$ 的值越大的个体质量越好, 值越小的个体质量越差的特性.

基于实数编码表示的基本定义和空车流量分配优化问题数学模型的特点, 我们选择了向量个体与优化问题解向量具有同样长度的实数编码表示方式, 既把空车流量分配优化模型的解向量 $f_i = (\dots, f_{ij}^k, \dots)^T = (f_{i1}, f_{i2}, \dots, f_{in})^T$ 设计为 RGA 应用算法的实数编码向量个体 x_i . 并设定对应于该实数编码向量个体的适应度函数有如下形式:

$$f_i = (Z(f_i)) = \begin{cases} \frac{1}{1 + (1 + 0.1)^{Z(f_i)}}, & \text{if } Z(f_i) \geq 0 \\ \frac{1}{1 + (1 - 0.1)^{Z(f_i)}}, & \text{if } Z(f_i) < 0 \end{cases}, \tag{3}$$

式中, α, β 为常数, 对于极小化问题常取 $\alpha = 1, \beta = \begin{cases} 1, & \text{if } Z(f_i) \geq 0 \\ -1, & \text{if } Z(f_i) < 0 \end{cases}$.

该适应度函数是空车流量分配多目标优化模型 [式 (2)] 中目标函数 $Z(f)$ 的等效度量函数, 可用来度

量无约束条件下解的质量,并且有 $\max (f_i)$ 等效于 $\min Z(f)$ 的特性,是评价个体以及设计 RGA 应用算法的基本依据。

由于 GA 算法只是通过目标函数与优化问题发生关系,即仅依据优化问题目标函数建立的适应度函数来评价个体,并未考虑约束条件对个体选择的影响。所以 GA 算法一般只可直接应用于求解无约束最优化问题。对于有约束的空车流量分配优化问题[式(2)],在选择以解向量 f 作为实数编码向量个体进行 RGA 应用算法设计时,首先必须考虑对模型约束条件的处理。根据约束条件 R 的不同特点可采用下列两种处理方法。

将模型约束条件转变为 RGA 个体分量边界值的处理方法

该处理方法的基本思想是:在 RGA 算法采用与解向量同样长的实数编码表示的条件下,个体中每个分量要在约束条件 R 规定的区域内进行选取,并且遗传算子在设计上应满足该约束条件。

设实数编码向量个体 $f_i = (\dots, f_{ij}^k, \dots)^T$ 的分量 f_{ij}^k 的取值区域为 $[f_{ij\min}^k, f_{ij\max}^k]$,其中 $f_{ij\min}^k$ 和 $f_{ij\max}^k$ 分别为个体分量的最小(下限)边界值和最大(上限)边界值。由模型的约束条件 $g_3(f)$ 和 $g_4(f)$ 可见,每个个体分量的取值必须大于等于零,最小边界值为 $f_{ij\min}^k = 0$;又由约束条件 $g_2(f)$ 和 $g_1(f)$ 可见,每个个体分量的最大边界值应不能超过 O-D 对 i, j 间的运量需求数 x_{ij} ,同时也不能大于路径 p_{ij}^k 上的运输能力 b_{ij}^k ($b_{ij}^k = \min\{b_e : e \in p_{ij}^k, \forall e \in E\}$),所以每个个体分量的最大(上限)边界值可取为 $f_{ij\max}^k = \min\{x_{ij}, b_{ij}^k\}$ 。将约束条件 R 规定的区域(变量取值的上下限值)转化为个体分量 f_{ij}^k 表示的最大值边界值 $f_{ij\max}^k$ 和最小边界值 $f_{ij\min}^k$,则 RGA 算法的初始化过程即可保证满足函数优化问题的所有约束条件,而且在 RGA 运行过程中,相应遗传算子也可以始终保持 f_{ij}^k 在其取值范围 $[f_{ij\min}^k, f_{ij\max}^k]$ 内选取。

该处理方法的不足之处是它只适用于 $l_{ij} = 1$ (既 O-D 对 i, j 间只有一条路径)的特殊情况。因为它只考虑了 O-D 对 i, j 间流经路径 p_{ij}^k 的部分流量 f_{ij}^k 不超过 i, j 间运量需求总量 x_{ij} 的情况,在 i, j 间具有多条合理路径时,并不能保证 i, j 间所有合理路径流量之和 $\left(\sum_{k=1}^{l_{ij}} f_{ij}^k\right)$ 小于等于需求总量 x_{ij} ,也就是不能完全处理约束条件 $g_2(f)$,同理也不能完全处理约束条件 $g_1(f)$ 。因此,应用该处理方法具有一定的局限性。

化有约束为等效无约束的处理方法

该处理方法的基本思想是:通过引入惩罚函数,对优化模型中不可行解进行惩罚,利用算法将有约束化为无约束问题,使问题的求解可直接采用 RGA 通用算法予以实现。

基于适应度函数式(3)与空车流量分配多目标优化模型式(2)中目标函数 $Z(f)$ 的等效性,我们设对应于空车流量分配优化模型约束条件 $g_1(f)$ 、 $g_2(f)$ 和 $g_4(f)$ 的惩罚函数有如下形式:

$$W(f) = 1/[(1 + 0.1)^{(|g_1(f)| + |g_2(f)| + |g_4(f)|)}], \quad (4)$$

由该惩罚函数式和式(3),可构造空车流量分配优化模型的等效无约束表达式为:

$$\max_{f \in R} \text{fit}(f) = Z(f) \cdot W(f). \quad (5)$$

显然该等效无约束表达式的解空间允许包含有约束模型[式(2)]的不可行解,在应用 RGA 算法求解时,由于惩罚函数的作用,对于不可行解将根据其违反约束的程度给以一定量的惩罚,违反约束程度越大的解,其在 RGA 算法中的适应度度量越小,如此随着 RGA 算法的运行,有约束模型[式(2)]的不可行解在解群中所占的比例将越来越小,而可行解将逐渐在群体中占据主导地位,并逐渐趋向于最好解。

该处理方法的特点是:在 RGA 算法采用与解向量同样长的实数编码表示的条件下,个体中每个分量不必再直接考虑受约束条件 R 的限制,可对约束条件 $g_2(f)$ 和 $g_1(f)$ 进行完全处理,并实现各遗传算子的操作也能满足约束条件,是一种更具有一般通用的处理方法。

由此所建立的等效目标函数 $\text{fit}(f)$ [式(5)],同时满足定义 3 中实数编码适应度函数的要求,将直接作为空车流量分配优化模型式(2)的适应度函数进行 RGA 算法设计,并由下式表示。

$$\text{fit}^*(f_i) = \text{fit}(f_i) \cdot W(f_i). \quad (6)$$

2.2 基于实数编码的遗传算子的设计

遗传算法中的遗传算子(或遗传策略)是实现复制(繁殖)、交叉和变异操作的重要工具,也是设计构造

遗传算法的主要内容.传统的遗传算子大都是针对二进制编码设计的,不适应于实数编码的形式,为此需要基于实数编码技术设计构造新的遗传算子.

1) 复制(繁殖 reproduction)算子的设计

常见的复制选择策略(复制算子)主要有排序策略、轮换策略和最优保存策略等多种形式^[5],其中轮换策略是一种实际应用最多的随机性选择,该策略存在的问题是对群体中适应值较差的个体有时也可能被选中,这将会影响到遗传算法的执行效果,为避免这种情况的发生,我们采用了轮换法与最优保存策略相结合的方法,称之为“轮换优先法策略”(roulette wheel-elitism method),既把轮换法产生的目前解群中最好的解(有最大适应值的解)直接放入到下一代解群中,保证加入匹配池中的串具有较大的适应性,使每代的解群中总会保留到目前为止所得到的最好解.由此设计构造了如下的实数编码轮换优先复制策略(算子).

定义4 复制(繁殖)算子 $S: \mu \rightarrow \mu$, 指在父代个体空间 μ 中, 根据实数编码向量个体 $f_i \in \mu$ 的适应度函数值 $f^*(f_i)$, 采用轮换优先法的选择策略, 以生存概率 $P_s(f_i)$ 实现复制个体 $f_i \in \mu$ 到匹配池中的操作.

设在一个群体中实数向量个体 f_i 的生存概率(选择概率或复制概率)为

$$P_s(f_i) = \frac{f^*(f_i)}{\sum_{j=1}^{\mu} f^*(f_j)}, \quad (7)$$

式中, $f^*(f_i)$ 为式(6)所示的空车流量分配优化模型的实数向量个体适应度函数值, $\sum_{j=1}^{\mu} f^*(f_j)$ 为整个实数群体的适应度值之和, $\mu - 1$ 为父代群体规模, 并设子代群体规模(通过重组和变异后产生的子代个体数目)与父代群体规模相等.

令: $s^* = \sum_{j=1}^{\mu} f^*(f_j)$, 则采用实数编码轮换优先法策略繁殖个体的基本步骤为:

Step1 求出父代群体中所有个体的适应值 $f^*(f_i)$;

Step2 从所有父代个体中选出适应值最大的个体优先加入匹配池;

Step3 求出父代群体中所有个体(不含适应值最大个体)的适应值之和 s^* ;

Step2 产生一个位于 $[0, s^*]$ 间均匀分布的伪随机数 r ;

Step3 从群体中编号为 1 的个体开始, 将其适应值与后继个体的适应值相加, 直到累加和等于或大于 r ;

Step4 将群体中满足下述条件的第 i 个个体加入匹配池:

$$\sum_{j=1}^{i-1} f^*(f_j) < r, \quad \text{同时} \quad \sum_{j=1}^i f^*(f_j) \geq r;$$

Step5 如果匹配池中包含的个体达到所需要的数目时, 则停止复制, 否则转 Step3.

该实数编码轮换优先复制选择的结果是返回一个随机的个体, 除适应值最大的个体优先被选择并保留之外, 每个个体被选择的机会将直接与适应值成比例, 那些没被选中的个体则从群体中被淘汰出去, 体现了“适者生存, 优胜劣汰”的复制原则.

2) 交叉(crossover)算子的设计

定义5 交叉算子 $C: \mu \rightarrow \mu$, 指在父代个体空间 μ 中, 以交叉概率 P_c 实现对任意两个实数向量个体 $f_a \in \mu, f_b \in \mu$ 进行断点交叉重组的操作.

常见的二进制交叉重组方法是先对个体随机选择交叉点, 再交换两个个体相应的部分. 根据实数编码的特点, 交叉算子应具有数值运算特征, 以利于实现两个实数向量个体间的信息交流和基因重组. 为此我们选择了两断点交叉方式, 并设计了实现交叉操作的线性组合算子. 交叉重组实现步骤如下:

Step1 从父代群体(父代个体空间 μ)中以交叉概率 P_c 随机选择两个实数编码个体, 设为:

$$f_a = x = (x_1, \dots, x_l)^T, \quad f_b = y = (y_1, \dots, y_l)^T;$$

Step2 对实数编码个体随机选定两个断点(交换点) i 和 j (i 和 j 为 $[1, l]$ 间的一致随机整数且 $i < j$), 并利用下列交叉操作线性组合算子计算断点间各交叉分量,

$$x_k = y_k * \alpha + x_k * (1 - \alpha), \quad y_k = x_k * \beta + y_k * (1 - \beta), \quad k = i + 1, \dots, j, \quad (8)$$

其中,参数 α 的取值为 $[0, 1]$ 间的常数;

Step3 进行交叉重组操作,产生两个子代实数编码向量个体:

$$\begin{aligned} f_a = x &= (x_1, \dots, x_i, x_{i+1}, x_{i+2}, \dots, x_{j-1}, x_j, x_{j+1}, \dots, x_l)^T, \\ f_b = y &= (y_1, \dots, y_i, y_{i+1}, y_{i+2}, \dots, y_{j-1}, y_j, y_{j+1}, \dots, y_l)^T. \end{aligned}$$

3) 变异 (mutation) 算子的设计

定义 6 变异算子 $m: I^\mu \rightarrow I^\mu$, 指演化到第 g 代时,在个体空间 I^μ 中,以变异概率 P_m 随机选中实数向量个体 $f_i \in I^\mu$ 的 f_{ik} 分量进行变异的操作.

针对实数编码表示的特点,本文设计了可随群体演化进程动态变化的数值变异算子,即个体 $f_i \in I^\mu$ 的分量 f_{ik} 经一次变异后的结果 f'_{ik} 由以下变异算子决定:

如果随机数为 1,则 $f'_{ik} = f_{ik} + r(g, R - f_{ik})$;

如果随机数为 0,则 $f'_{ik} = f_{ik} - r(g, f_{ik} - L)$.

式中, L 和 R 分别是 f_{ik} 的上下边界; r 是 $[0, 1]$ 上的随机数; (g, x) 为漂移函数,定义为

$$(g, x) = \left[1 - e^{-\left(1 - \frac{g}{G}\right)^k} \right] x, \quad (9)$$

式中, g 是当前遗传代数, G 是总的遗传代数, x 是实数 ($R - f_{ik}$ 或 $f_{ik} - L$), k 是漂移参数,它决定了漂移函数在 $(0, x)$ 上漂移的程度. 漂移函数 (g, x) 返回 $(0, x)$ 上的一个值,该值随着遗传代数 g 的增加逐渐趋向于 0,使得搜索领域能动态地被调整,实现动态漂移性变异. 既一方面可实现从当前解漂移到解空间的另一部分,有助于跳出局部最优,保持群体的多样性;另一方面可实现在群体演化的初期阶段大幅度漂移变化,以很大的概率找到全局最优解,当演化进行到后阶段时搜索领域范围自动减少,进入局部搜索,使算法较快地收敛到近似最优解.

4) 爬山 (hill-climbing) 算子的设计

定义 7 爬山算子 $h: I^\mu \rightarrow I^\mu$, 指以概率 P_h 随机从个体空间 I^μ 中选择 h ($h \leq \mu$) 个实数向量个体 $f_i \in I^\mu, i = 1, 2, \dots, h$, 实行局部寻优的爬山搜索操作.

爬山算子是对实数编码遗传算法的一个补充算子,设置该算子的目的主要是提高算法的收敛速度和品质^[6,7],只在算法收敛减慢时开始采用.在算法中,爬山算子是与复制算子相结合使用的,即爬山算子只对父代群体中 h 个实数向量个体实行爬山搜索操作,剩余的 $\mu - h$ 个个体则根据复制算子实行复制操作,两个算子作用的结果合并构成一个新的子代群体.爬山算子的实现步骤为:

Step1 以概率 P_h 随机从群体中选择 h 个个体 $f_i \in I^\mu, i = 1, 2, \dots, h$ 进行局部爬山搜索;

Step2 按目标函数值 $\text{fit}(f_i)$ [式(5)]的大小评价 h 个个体的优劣,设最差个体点为 f_p ;

Step3 计算除最差个体点外的 $h - 1$ 个顶点个体的形心 fit 及 f_p 关于形心的反射点 F :

$$\text{fit} = \frac{1}{h-1} \left(\sum_{i=1}^h f_i - f_p \right), \quad F = \text{fit} + (\text{fit} - f_p), \quad (\text{式中 } \alpha \text{ 为反射系数}). \quad (10)$$

Step4 若反射点 F 的目标函数值好于 h 个顶点个体中的次差点,则以 F 取代 f_p ,作为新的子代个体,结束爬山过程,否则,转入下步;

Step5 调整反射点向形心收缩: $F = (\text{fit} + F)/2$,返回 Step4.

在实际算法中,一般启动爬山算子的条件是:当前演化代 (g 代) 中目标函数的最好值 $\text{fit}(f_i^g)$ 与上一代 ($g - 1$ 代) 中目标函数的最好值 $\text{fit}(f_i^{g-1})$ 之差小于一给定常数 $\epsilon > 0$.

2.3 空车流量分配优化问题 RGA 算法的基本构成及特点

上述基于实数编码设计的各遗传算子 s, c, m, h 均为作用于个体上的微算子,其相应的宏算子,可相应地记为 S, C, M, H ,分别表示作用在群体上的复制算子、交叉算子、动态变异算子和爬山算子,这些宏算子是构成空车流量分配优化问题遗传算法的基本元素.

设: $\{T, F\}$ 表示遗传算法停止的判断准则, T 表示停止准则为真, F 表示停止准则为假;

P^0 表示初始群体; P^g 表示第 g 代群体; μ 表示群体规模;

$\{T, F\}$ 表示爬山算子的启动准则, T 表示启动爬山算子, F 表示不启动爬山算子.

基于上述符号, 可把基于实数编码方式求解空车流量分配优化问题的遗传算法描述为: $RGA = (P^0, \mu, S, C, M, H, \dots)$, 该算法的基本构成及步骤可用伪码描述为下列形式.

Begin

$g := g + 1;$

initialization $P^0 := \{f_1^0, \dots, f_\mu^0\}$; //初始化过程, 随机产生初始种群

evaluation $P^0 := \{f_1^0, \dots, f_\mu^0\}$; //计算初始种群的适应度函数值

While $(P^g) T$ do $g := g + 1$; //算法停止判断准则

selection $P^g := S(P^{g-1})$; //对 $g-1$ 代种群执行复制(繁殖算子)操作

crossover $P^g := C(P^g)$; //对 g 代种群执行交叉(算子)操作

mutation $P^g := M(P^g)$; //对 g 代种群执行变异(算子)操作

if $(P^g) = T$ //爬山算子启动准则

then hill-climbing $P^g := H(P^g)$; //对 g 代种群执行爬山(算子)操作

evaluation $P^g := \{f_1^g, \dots, f_\mu^g\}$; //计算 g 代种群的适应度函数值

end

该 RGA 算法的执行过程是: 首先由初始化过程 initialization 随机产生一组优化模型的初始可行解 f_i^0 ($i = 1, 2, \dots, \mu$), 每个初始可行解即为一个初始个体, 由 μ 个初始个体构成算法的初始种群 $P^0 = \{f_1^0, \dots, f_\mu^0\}$, 并根据优化问题的目标函数计算个体适应度函数, 构成初始种群适应度函数值集合 $\{f_1^0, \dots, f_\mu^0\}$; 然后基于该初始种群和初始种群适应度函数值集合, 运用复制算子 selection 选择下一代父辈种群的个体, 再经过交叉 crossover、变异 mutation、爬山 then hill-climbing 等算子操作产生新一代的种群, 该进化过程重复进行, 直到满足算法停止准则时为止.

该算法的特点主要表现为: 一是通用性强, 即算法的迭代过程是以优化变量的实数编码作为运算对象, 将变量编码成“染色体”后进行进化操作, 而不是直接依靠优化变量本身的实际值进行优化运算, 对不同优化问题的求解, 只需改变优化变量编码便可应用同一算法实现, 不但可避免对连续函数变量编码与解码的过程, 还可避免算法对函数优化问题本身复杂性的依赖. 二是可避免大量复杂计算, 减少优化问题求解计算上的难度, 即算法仅使用由目标函数值变换来的适应度函数值作为搜索信息, 便可确定进一步的搜索方向和搜索范围, 无需象传统的优化算法那样需要目标函数的导数等其他更多的辅助信息. 三是具有较强的鲁棒性, 即算法的最终解不依赖于初始解的选取, 原则上可在实数向量个体的每个分量的取值范围内随机产生任意的初始解, 然后通过 RGA 每个演化代中各遗传算子的作用, 逐步调整各个合理路径上的流量分布, 使之达到系统整体最优原则下的空车流量分布状态. 四是可大大提高搜索全局最优解的速度, 即在充分运用遗传算法固有的始群信息并行处理特有功效的同时, 还将群体演化过程与函数优化爬山搜索技术有机结合, 综合考虑了传统 GA 的随机全局寻优和爬山算法局部快速搜索的特点, 不仅可大大改善收敛速度, 而且还可改善解的品质. 五是收敛性好, RGA 算法是一种自适应概率搜索技术方法, 其选择、交叉、变异等运算都是以一种概率的方式来进行的, 并且在复制和爬山算法中采用了换代保优的处理方法, 可实现以概率 1 收敛于问题的最优解^[2].

3 应用 RGA 算法的实例分析

设铁路空车流量分配的路网结构如图 1 所示. 该路网由 22 个节点、32 条双向边构成, 其中节点集合为 $V = \{1, 2, \dots, 22\}$; 相邻节点间的每一条边代表两条对向弧(正向弧 $e_{\text{正}} = (i, j)$ 和反向弧 $e_{\text{反}} = (j, i)$), 弧集合为 $E = \{e = \{e_{\text{正}} = (i, j), e_{\text{反}} = (j, i)\} \mid i, j \in V, e = 1, 2, \dots, 64\}$, 集合中有分配流量弧的弧长度和弧上的运输能力容量值分别为表 2 中的 l_e 和 b_e . 已知空车需求量的 O-D 对及相应需求量 x_{ij} 、分配优先权系数 w_{ij} 的取值如表 3 所示(为了检验 RGA 算法的有效性, 在模拟分析实例中特意选择了一些可发生对流运输

的空车需求 O-D 对)。

通过专家咨询定量评判分析,确定空车流量分配优化模型[式(2)]的目标函数 Z_1 (总运输耗费最少) Z_2 (总分配流量最大)和 Z_3 (总社会效用最大)的线性加权和系数分别取值为

$\alpha_1 = 0.5, \alpha_2 = 0.3, \alpha_3 = 0.2$, 且 $\alpha_1 + \alpha_2 + \alpha_3 = 1$. 实例分析选择 RGA 算法的有关控制参数为:群体规模 $\mu = 50$, 交叉概率 $p_c = 0.6$, 变异概率 $p_m = 0.05$. 爬山概率用于控制爬山搜索操作的频率, 概率越大爬山搜索效果越显著, 但相对会增加计算量, 实际中可结合爬山算子启动准则综合考虑. 本例取爬山概率 $p_h = 1$, 参与局部爬山搜索的个体数 $h = 50$, 爬山算子的启动条件 $\text{差}(f_i^g) - \text{差}(f_i^{g-1}) = 0.01$, 总遗传代数 $G = 50$.

表 1 中给出了采用 RGA 算法计算的各路段弧流量 f_e 及饱和度的结果(未列入表中的弧的弧流量均为零), 由各计算值可见, 各路段弧流量均能满足模型关于运输能力约束的条件要求, 并且体现了尽可能充分地利用路网能力的优化目标, 验证了 RGA 算法对能力约束条件处理的有效性。

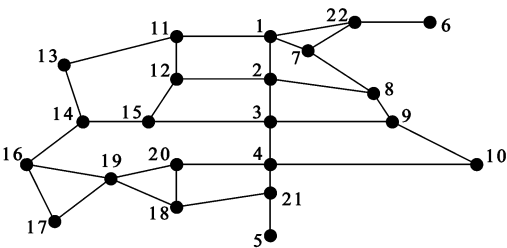


图 1 路网结构示意图

表 1 路段弧的有关参数及弧流量分配情况

编号	弧 $e = (i, j)$	弧长度 l_e	弧能力 b_e	弧流量 f_e	饱和度 %
1	(22,1)	299	350	99	28.3
2	(22,7)	265	392	380	97
3	(7,1)	137	385	380	98.7
4	(8,9)	317	230	115	50
5	(9,10)	649	290	167	73.7
6	(2,8)	301	169	115	68.4
7	(3,9)	349	55	52	94.5
8	(4,10)	1148	24	15	62.1
9	(1,2)	283	900	894	99.3
10	(2,3)	412	388	250	64.4
11	(3,4)	945	295	250	84.8
12	(4,21)	134	319	304	95.4
13	(11,12)	282	412	300	72.8
14	(2,12)	231	108	50	47.5
15	(15,3)	511	67	52	77.6
16	(12,15)	651	98	50	52.3
17	(11,13)	1429	86	77	89.5
18	(14,15)	153	61	52	85.2
19	(16,14)	669	45	12	26.7
20	(19,20)	458	58	54	93.6
21	(20,4)	447	63	54	86.2
22	(19,18)	610	23	5	21.7
23	(18,21)	538	25	5	20
24	(21,5)	521	311	309	99.4
25	(6,22)	989	486	479	98.6

表 2 中给出了采用 RGA 算法计算的空车流量分配最优路径方案的结果. 为了进行比较分析, 表中列出了两种不同约束条件下的计算方案. 一种是允许对流运输条件下的方案, 即去掉式(4)惩罚函数中无对流运输约束条件 $g_4(f)$ 的最优路径计算方案, 共由 17 条路径构成, 分别为: $p_{1,5}^1, p_{1,10}^1, p_{1,10}^2, p_{1,15}^1, p_{1,15}^2, p_{1,17}^1,$

$p_{1,17}^2, p_{1,17}^3, p_{4,10}^1, p_{6,2}^1, p_{6,2}^2, p_{11,12}^1, p_{11,13}^1, p_{14,10}^1, p_{16,10}^1, p_{19,5}^1, p_{19,5}^2$, 各路径的分配流量 f_{ij}^k 为表中括号内的值, 总的分配流量为: $f_{1,5}^1 + f_{1,10}^1 + f_{1,10}^2 + f_{1,15}^1 + f_{1,15}^2 + f_{1,17}^1 + f_{1,17}^2 + f_{1,17}^3 + f_{4,10}^1 + f_{6,2}^1 + f_{6,2}^2 + f_{11,12}^1 + f_{11,13}^1 + f_{14,10}^1 + f_{16,10}^1 + f_{19,5}^1 + f_{19,5}^2 = 1593$, 可满足 $(1593/1640 =) 97\%$ 空车 O-D 量的运输需求, 该方案中存在对流运输的双向弧分别为 (1, 7)、(3, 15)、(14, 15)、(4, 20)、(19, 20). 另一种是考虑无对流运输约束条件 $g_4(f)$ 的计算方案, 该方案中不存在双向弧的对流运输, 共有 12 条运输路径, 分别为 $p_{1,5}^1, p_{1,10}^2, p_{1,15}^1, p_{4,10}^1, p_{6,2}^1, p_{6,2}^2, p_{11,12}^1, p_{11,13}^1, p_{14,10}^1, p_{16,10}^1, p_{19,5}^1, p_{19,5}^2$, 总的分配流量为 $f_{1,5}^1 + f_{1,10}^2 + f_{1,15}^1 + f_{4,10}^1 + f_{6,2}^1 + f_{6,2}^2 + f_{11,12}^1 + f_{11,13}^1 + f_{14,10}^1 + f_{16,10}^1 + f_{19,5}^1 + f_{19,5}^2 = 1397$, 可满足 $(1397/1640 =) 85\%$ 空车 O-D 量的运输需求.

对比上述两种流量分配路径方案可见, 无对流方案的各条路径均包含在允许对流方案之中, 并且是允许对流方案中不含对流弧的路径, 或者是含对流弧但分配流量最大的路径, 例如允许对流方案中的路径 $p_{1,10}^1$ 和 $p_{6,2}^2$ 分别包含同一路段的双向弧 $e_{正} = (1, 7)$ 和 $e_{反} = (7, 1)$, 属于含对流弧的路径, 路径分配流量分别为 $f_{1,10}^1 = 114$ 和 $f_{6,2}^2 = 380$, 其中分配流量最大的路径 $p_{6,2}^2$ 正是无对流方案中的一条路径, 而分配流量较小的路径 $p_{1,10}^1$ 则被拒绝(在无对流优化中分配的流量为 $f_{1,10}^1 = 0$), 其他无对流方案中的路径 $p_{14,10}^1, p_{16,10}^1, p_{19,5}^1$ 也均属这种情况. 这一结论表明, RGA 算法对无对流运输约束条件下空车流量分配优化问题的求解是有效的, 并且充分体现了最大流空车分配的目标要求. 另外, 从表 2 中各 O-D 对间的流量分配比例来看, 显然允许对流方案满足空车 O-D 量的运输需求程度要比无对流方案好, 而且均满足 O-D 对间的分配流量不大于运输供求量的约束条件, 这都验证了采用 RGA 算法求解复杂空车流量分配优化问题的可行性和有效性.

表 2 空车 O-D 有关参数及空车流量分配方案

O-D 对 (i, j)	空车需求量 x_{ij}	分配优先权 w_{ij}	分配总量/分配比例 %	无对流(允许对流) f_{ij}^k	路径 p_{ij}^k
1, 5	250	0.2	250/100	250(250)	1 2 3 4 21 5
1, 10	230	0.15	115/50 (229)/(99.6)	0(114)	<u>1</u> 7 8 9 10
				115(115)	1 2 8 9 10
1, 15	100	0.1	50/50 (96)/(96)	50(50)	1 2 12 15
				0(46)	1 2 <u>3</u> 15
1, 17	50	0.05	0/0 (36)/(72)	0(2)	1 2 12 15 <u>14</u> 16 17
				0(4)	1 2 <u>3</u> 15 14 16 17
				0(30)	1 2 3 <u>4</u> 20 19 17
4, 10	20	0.1	15/75	15(15)	4 10
6, 2	480	0.1	479/99.8	99(99)	6 22 1 2
				380(380)	6 22 <u>7</u> 1 2
11, 12	300	0.1	300/100	300(300)	11 12
11, 13	80	0.05	77/96.3	77(77)	11 13
14, 10	50	0.05	40/80	40(40)	<u>14</u> 15 <u>3</u> 9 10
16, 10	20	0.05	12/60	12(12)	16 <u>14</u> 15 3 9 10
19, 5	60	0.05	59/98.3	54(54)	<u>19</u> 20 <u>4</u> 21 5
				5(5)	19 18 21 5

注: 表中带下划线的路段为路径中存在对流的路段弧.

4 结束语

本文针对空车调整中流量分配非线性模型求解计算的复杂性, 提出了采用实数编码遗传算法求解的新方法. 该求解方法具有不直接依赖于模型本身结构的明显特点, 避免了采用常规优化技术在计算和算法构造上的困难, 是一种容易实现、可操作性强, 比较实用的方法. 该方法已被应用于铁路空车调整计划分析的实际, 并取得了很好的效果.

(下转第 144 页)

实现传统故障诊断规则的改进,数据来源于飞机的设计和使用全过程,在诊断中更有效、实用。

参考文献:

- [1] 徐荣红,孙金标.基于神经网络和专家系统的电传动操纵系统故障诊断[J].航空学报,2005,26(2):195-198.
Xu Ronghong,Sun Jinbiao. Fault diagnosis for the fly-by-wire system based on artificial neural network and expert system[J]. Acta Aeronautica Et Astronautica Sinica,2005,26(2):195-198.
- [2] 金林,张洪才.一种基于模糊神经网络的故障诊断方法[J].西北工业大学学报,2004,22(5):657-661.
Jin Lin,Zhang Hongcai. On a new method of intelligent fault diagnosis[J]. Journal of Northwestern Polytechnical University,2004,22(5):657-661.
- [3] 董健康,耿宏.飞机排故方法模糊与统计综合评估算法的研究[J].航空学报,2004,25(3):258-262.
Dong Jiankang,Geng Hong. Research on fuzzy and statistical algorithms of comprehensive evaluation methods to eliminate airplane faults[J]. Acta Aeronautica Et Astronautica Sinica,2004,25(3):258-262.
- [4] Air Transport Association of America, Inc. 20004-1707 ATA2200[S]. Washington DC ATA Inc,2003.
- [5] Bombardier. CSP A-001 CRJ200/440 Aircraft Maintenance Manual[Z]. Canada,2002:1053-1064.
- [6] Bombardier. CSP B-053 CRJ700/CRJ900 Maintenance Review Board Report[Z]. Canada,2001:58-70.
- [7] Bombardier. CSP A-113 CRJ100/200 Quarterly/Monthly FRACAS report[Z]. Canada,2004:88-89,135-136.
- [8] 杨树子,等.基于知识的诊断推理[M].北京:清华大学出版社,1993:5-15.
Yang Shuzi,et al. Diagnosis Reasoning Based on Knowledge[M]. Beijing: Tsinghua University Press,1993:5-15.
- [9] 刘清. Rough集及 Rough推理[M].北京:科学出版社,2001:13-43.
Liu Qing. Rough Set and Rough reasoning[M]. Beijing: Science Press,2001:13-43.
- [10] 蒙祖强,蔡自兴.个性化决策规则的发现:一种基于 Rough Set的方法[J].控制与决策,2004,19(9):994-998.
Meng Zuqiang,Cai Zixing. Discovery of personalized decision rule: A rough set-based approach[J]. Control and Decision,2004,19(9):994-998.

(上接第138页)

参考文献:

- [1] 林柏梁,等.铁路上带权重的车流径路优化方法[J].北方交通大学学报,1996,20(6):645-650.
Lin Boliang,et al. Optimal weighted car routing in railway network[J]. Journal of Northern Jiaotong University,1996,20(6):645-650.
- [2] 王凌.智能优化算法及其应用[M].北京:清华大学出版社,2001.
Wang Ling. Intelligent Optimization Algorithms with Applications[M]. Beijing:Tsinghua University Publishing house,2001.
- [3] 张喜.铁路空车调整综合优化模型与智能决策方法的研究[D].北京交通大学,2004.
Zhang Xi. Study on Optimal Model and Intelligent Decision Method of Railway Empty Wagons Distribution[D]. Beijing Jiaotong University,2004.
- [4] 周全荣.遗传算法的改进及其应用研究[J].控制与决策,1995,10(3):40-43.
Zhou Quanrong. Improved genetic algorithm with applications[J]. Control and Decision,1995,10(3):40-43.
- [5] Goldberg D. Genetic Algorithm in Search,Optimization and Machine Learning[M]. Addison Wesley,1989.
- [6] Mublenbein H,Gorges-schleuter M,Kramer O. Evolution algorithms in combinatorial optimization[J]. Parallel Computing,1988,7:65-85.
- [7] Mublenbein H,M. How Genetic Algorithms Really Work I. Mutation and Hill-climbing[G]//R. Manner and B. Manderick. Parallel Problem Solving form Nature,1992,2:15-25.