幻方问题的演化算法*

谢 涛[1] 赵 彬[1] 谢道裕[2]

[1](国防科学技术大学 计算机学院 长沙 410073)
[2](湖南天演科技有限公司 智能计算与控制中心 长沙 410013)

摘 要 幻方问题是具有悠久历史的复杂排列组合问题.幻方问题的复杂性不仅在于解的多样性随阶数指数递增,而且在于解在可行排列空间中所占的比例随阶数指数递减.本文在提出半幻方通过行置换与列置换可实现对角线数字幻和满足的分步构造猜想的基础上,提出基于演化策略的分步自适应幻方演化算法.变异操作包括元素对置换、整行置换、整列置换;启发式局部调整操作包括行列局部调整与对角局部调整等.计算表明,分步构造猜想至少在所完成的幻方构造计算实例上是成立的,幻方分步演化算法具有较高的计算效率.

关键词 幻方,演化算法,分步构造猜想
中图法分类号 TP301

Evolutionary Algorithm for Magic Squares

XIE Tao[1], ZHAO Bin[1], XIE Dao-Yu[2]
(*College of Computer, National University of Defense Technology, Changsha* 410073)
(*Intelligence Computations and Control Center, Hunan Evonature Science and Technology Corporation, Limited, Changsha* 410013)

ABSTRACT

Magic square construction is a complex permutation problem with a long history. The complexity not only consists of the number of magic squares that increases rapidly with the order of magic square, but also of the percentage of magic squares in the possible permutation of the first n^2 natural numbers that decreases with the order. Based on the two-phase construction conjecture, an improved evolutionary algorithm for magic square construction is proposed. Mutation operators are specially designed so that the mutation domain can be located and the mutation probabilities can be adjusted adaptively which include the number transpositions, the row transpositions and column transpositions. In addition, some heuristics-based local permutations, such as the local row/column rectification and the local diagonal rectification, are used to complement the stochastic mechanism. Computational results show that the two-phase construction conjecture is computationally effective, and the improved evolutionary algorithm is highly efficient for magic square construction.

Key Words Magic Square, Evolutionary Algorithms, Two-Phase Construction Conjecture

* 国家自然科学基金资助项目(No. NSF60473011, NSF60133010)
收稿日期:2005-08-23;修回日期:2005-12-05
作者简介 **谢涛**,男,1966 年生,教授,博士后,主要研究方向为软计算、组合数学、密码学、网络信息安全、复杂性与复杂科学. E-mail: taoxie@nudt.edu.cn. **赵彬**,男,1980 年生,硕士,主要研究方向为智能控制与网络信息安全. **谢道裕**,男,1973 年生,工程师,主要研究方向为智能控制与信息系统集成.

1 引　言

将 n^2 个最小连续自然数填入 $n\times n$ 的矩阵中，使得每行元素之和、每列元素之和以及对角线元素之和均等于同一常数，满足以上条件的矩阵称为幻方.幻方最早源于公元前 23 世纪大禹治水时代的“河图洛书”传说，在中国汉朝与宋朝分别称为九宫图与纵横图.幻方在人工智能、图论、对策论、实验设计、工艺美术、电子回路原理、集成电路设计、位置解析学等方面有着潜在的广泛应用.

8	1	6
3	5	7
4	9	2

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

(a)中国洛书　　(b)欧洲的 Durer 幻方

(a)Chinese Luoshu　　(b)Durer's magic squares

图 1　幻方

Fig. 1　Magic squares

即使不计由幻方旋转或反射变换所产生的数量，幻方的数量也是十分巨大的，而且随阶数指数增长.虽然 Bessy 早在 17 世纪就列出 4 阶幻方的全部 880 种排列方式[1-2]，最近又有人计算 5 阶幻方的数量共有 275 305 224 种，但 6 阶幻方的数量至今未有准确结果. Pinn 与 Wieczerkowski[3] 应用 Monte Carlo 仿真方法，估计 6 阶幻方数量高达 $(1.7745\pm 0.0016)\times 10^{19}$. 然而，确定任一阶数幻方的数量是目前仍未解决的难题[4].

Kraitchik[5] 在 1942 年分别给出奇数阶与偶数阶幻方的确定构造方法，但却不能构造任意随机幻方，更不能构造有附加条件或二次以上的幻方.在幻方研究中常常需要构造具有附加条件的特殊幻方，如泛幻方(panmagic square)、嵌套幻方(父子幻方)、庆典幻方等，每一个成功的特殊幻方的直接构造都是一次人类心智与毅力的艰苦磨砺，有时虽耗费一生光阴也一无所得.

与直接构造思想本质不同的是，本文基于解空间搜索的思想，把 n 阶幻方的构造过程看成是对 n^2 个连续自然数的可能排列空间的一个搜索过程.这样，幻方的构造就可以转化为一般的组合优化问题，采用组合优化算法求解.幻方的数量随着阶数的递增呈指数增长，并不意味着阶数越高幻方构造越容易.恰恰相反，根据初步统计结果，阶数每递增一阶，相应幻方数量在所有可能的排列矩阵中的比例下降百万倍，这意味着幻方的机器搜索难度随着阶数递增而呈指数级增长.计算技术的发展给幻方的研究与应用带来新的希望，虽然人类可以借助计算机解决非常复杂的工程数值计算问题，但复杂组合优化问题(NP-难)的求解效率并非依赖机器的速度.复杂组合优化问题需要高效的组合优化算法.进化计算可用于 TSP 等问题的求解，是一种可成功用于求解复杂组合优化问题的自适应智能算法，但基于进化计算的幻方随机演化构造算法却未见文献发表[6].本文基于进化计算中的演化策略[7]，探索随机幻方构造问题的高效演化算法.

2 随机幻方构造的简单演化算法

有关幻方问题的随机构造方法至今未见学术文献正式报道.与 TSP 问题一样，幻方问题是一个复杂的排列组合问题，不同排列组合之间的杂交可能生成非法个体，因而幻方演化算法中不使用杂交或重组算子.杂交算子是遗传算法的主要算子，除去杂交算子的遗传算法近似随机搜索算法.相反，演化策略中的重组算子只是次要算子，而变异算子才是主要算子.因此，随机幻方构造的演化算法一般采用演化策略.

进一步，设矩阵 $\boldsymbol{M}=(a_{ij})_{n\times n}$，其中 $a_{ij}\in\{1,2,3,\cdots,n^2\}$，且 $a_{ij}\neq a_{kl}, i\neq k$ 或 $j\neq l$. 如果满足

$$\sum_{i=1}^{n}a_{ij}=c,\ \sum_{j=1}^{n}a_{ij}=c,\ \sum_{i=1}^{n}a_{ii}=c,\ \sum_{i=1}^{n}a_{i,n-i+1}=c,$$

其中

$$c=\frac{1}{2}[n(n^2+1)],\ i,j=1,2,\cdots,n;$$

则称矩阵 $\boldsymbol{M}$ 为 n 阶幻方，相应常数 c 称为 n 阶幻方的幻和或幻数.

定义亲本集合 U，子代集合 Θ，种群集合 $\Pi=U\cup\Theta$，并且设 $|U|=\mu$，$|\Theta|=\lambda$，其中 $|U|$ 和 $|\Theta|$ 分别表示集合 U 与集合 Θ 中的个体数目.简单演化算法一般采用 $(\mu+\lambda)-ES$ 更新机制.

编码表示　采用演化策略的二层表示方法，$I=(\boldsymbol{M},\boldsymbol{\Delta})$，其中 $\boldsymbol{M}=(a_{ij})_{n\times n}$，$\boldsymbol{\Delta}=(\sigma_{ij})_{n\times n}$，$\sigma_{ij}$ 为相应 a_{ij} 的变异方差；一般初始设置 $\sigma_{ij}=3, i,j=1,2,\cdots,n$.

变异操作　设亲本个体 $I=(\boldsymbol{M},\boldsymbol{\Delta})$，变异后代个体 $I'=(\boldsymbol{M}',\boldsymbol{\Delta}')$，每一元素以一定变异概率 p_m 被选为变异对象，定义 $rand(x,y)$ 为 $[x,y]$ 内均匀随机整数产生函数.假定元素 a_{ij} 被选为变异对象，则 $a'_{ij}=$

$a_{ij}+rand(-\sigma_{ij},\sigma_{ij})$. 如果 $a'_{ij}<1$,置 $a'_{ij}=rand(1,n)$;如果 $a'_{ij}>n^2$,置 $a'_{ij}=n^2-rand(0,n)$. 设 $a_{kl}=a'_{ij}$,则置 $a'_{kl}=a_{ij}$,即将 M 中两元素 a_{ij} 与 a_{kl} 互换. $\sigma'_{ij}=\sigma_{ij}+rand(-1,1)$,如果 $\sigma'_{ij}<1$,则置 $\sigma'_{ij}=1$.

适应值函数　一般取每行、每列与两对角线上元素和与幻和 c 之差的绝对值的总和为目标函数值,即对于 $\boldsymbol{M}=(a_{ij})_{n\times n}$,

$$J(\boldsymbol{M})=\sum_{i=1}^{n}\left|c-\sum_{k=1}^{n}a_{ik}\right|+\sum_{j=1}^{n}\left|c-\sum_{k=1}^{n}a_{kj}\right|+\left|c-\sum_{i=1}^{n}a_{ii}\right|+\left|c-\sum_{i=1}^{n}a_{i,n-i+1}\right|.$$

设当前亲本中最大目标函数值为 $J'_{\max}$,则个体 $I=(\boldsymbol{M},\boldsymbol{\Delta})$ 的适应值为 $f(I)=J'_{\max}-J(\boldsymbol{M})$.

简单演化算法难以构造 10 阶以上幻方,而且演化效率极低.

3　幻方问题的改进演化算法

幻方分步构造猜想　一个行与列均满足幻和 c 的半幻方矩阵 $\boldsymbol{M}=(a_{ij})_{n\times n}$,仅通过有限次行置换和/或有限次列置换可使主对角线上元素满足幻和.

该猜想即是改进算法的基本原理. 行置换与列置换不改变行列元素之和,因此,算法可以首先找到行列满足的半幻方,然后通过行置换与列置换实现两对角线元素的幻和.

3.1　适应值函数设计

step1　半幻方演化过程的目标函数取为

$$J_1(\boldsymbol{M})=\sum_{i=1}^{n}\left|c-\sum_{k=1}^{n}a_{ik}\right|+\sum_{j=1}^{n}\left|c-\sum_{k=1}^{n}a_{kj}\right|,$$

其中,$n_{\text{row}}+n_{\text{col}}>0$,$n_{\text{row}}$ 与 n_{col} 分别为矩阵 $\boldsymbol{M}$ 中不满足幻和的行数与列数.

step2　半幻方的对角幻和演化过程的目标函数取为

$$J_2(\boldsymbol{M})=\left|c-\sum_{i=1}^{n}a_{ii}\right|+\left|c-\sum_{i=1}^{n}a_{i,n-i+1}\right|,$$

其中 $n_{\text{row}}+n_{\text{col}}=0$.

如果正对角线元素满足幻和,即 $\sum_{i=1}^{n}a_{ii}=c$,则置 $d_1=0$,否则置 $d_1=1$;同样,如果斜对角线元素满足幻和,即 $\sum_{i=1}^{n}a_{i,n-i+1}=c$,则置 $d_2=0$,否则置 $d_2=1$.

特别地,在 step1 构造过程中,当个体的所有行与所有列均满足幻和时,即 $n_{\text{row}}+n_{\text{col}}=0$ 时,算法转入 step2 演化过程. 分步演化可以实现行列幻和演化与对角幻和演化之间的解耦.

3.2　变异操作

设亲本个体 $I(\boldsymbol{M},\boldsymbol{\Delta})$,变异后代个体 $I'=(\boldsymbol{M}',\boldsymbol{\Delta}')$.

元素置换算子　自适应选择某些元素变异,然后与其相同或相近元素置换. 变异元素选择范围为未满足幻和的行列,分以下 3 种情况.

1) 变异对象为所在行列均不满足幻和的元素集合

$$S_1=\{a_{ij}:\sum_{k=1}^{n}a_{ik}\neq c,\sum_{k=1}^{n}a_{kj}\neq c,1\leqslant i\leqslant n,1\leqslant j\leqslant n\}.$$

交换范围为未满足幻和的行与列元素,

$$S_2=\{a_{i\cdot}:\sum_{k=1}^{n}a_{ik}\neq c,1\leqslant i\leqslant n\}\cup\{a_{\cdot j}:\sum_{k=1}^{n}a_{kj}\neq c,1\leqslant j\leqslant n\}.$$

因为元素变异值不一定可在 S_2 中找到对应的等值元素,因此可以在 S_2 中与元素变异值相差最小的元素互换. 如图 2 所示数字矩阵(7×7),假设阴影网格单元所在行(2,4,5 行)与列(3,5 列)表示未满足幻和的行与列,则其交叉部分即图中黑色单元表示元素集合 S_1,而阴影网格单元与黑色单元一起构成元素集合 S_2.

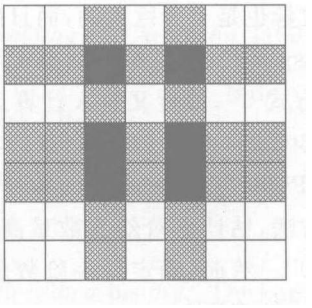

图 2　元素集合 S_1 与 S_2 图例

Fig. 2　Illustration of sets S_1 and S_2

变异概率 $p_m=1/(n_{\text{row}}n_{\text{col}})$,即可变异对象集合 S_1 中平均仅发生一次变异. 假定元素 $a_{ij}\in S_1$ 被选为变异对象,则有 $a'_{ij}=a_{ij}+rand(-\sigma_{ij},\sigma_{ij})$. 如果 $a'_{ij}<1$,置 $a'_{ij}=rand(1,n)$;如果 $a'_{ij}>n^2$,置 $a'_{ij}=n^2-rand(0,n)$. 设 $a_{kl}=\min\limits_{a_{ij}\in S_2}|a'_{ij}-a_{ij}|$,则置 $a'_{kl}=a_{ij}$,即将 $\boldsymbol{M}$ 中两元素 a_{ij} 与 a_{kl} 互换. $\sigma'_{ij}=\sigma_{ij}+rand(-1,1)$. 如果 $\sigma'_{ij}<1$ 或 $\sigma'_{ij}>\sigma_t$,$\sigma'_{ij}=rand(1,\sigma_t)$. 此中

$$\sigma_t=\begin{cases}\dfrac{1}{n_{\text{row}}+n_{\text{col}}}\left(\sum\limits_{i=1}^{n}\left|c-\sum\limits_{j=1}^{n}a_{ij}\right|+\sum\limits_{j=1}^{n}\left|c-\sum\limits_{i=1}^{n}a_{ij}\right|\right), & \text{if } n_{\text{row}}+n_{\text{col}}\neq0\\ \dfrac{1}{d_1+d_2}\left(\left|c-\sum\limits_{i=1}^{n}a_{ii}\right|+\left|c-\sum\limits_{i=1}^{n}a_{i,n-i+1}\right|\right), & \text{if } n_{\text{row}}+n_{\text{col}}=0\end{cases}$$

2) 变异对象为未满足幻和的行与列元素集合 S_2,交换范围也为未满足幻和的行与列元素集合 S_2. 因为元素变异值不一定可在 S_2 中找到对应的等值元素,因此可以在 S_2 中与元素变异值相差最小的元素互换. 行变异概率 $p_m = 1/(2nn_{\text{row}})$,列变异概率 $p_m = 1/(2nn_{\text{col}})$,即可变异对象中平均仅发生一次变异. 除变异对象元素集合由 S_1 变为 S_2 外,其他操作与上一情况类似.

3) 变异对象为未满足幻和的行与列元素集合 S_2,交换范围为矩阵中所有元素. 因此,元素变异值一定可找到对应的等值元素进行互换. 行变异概率 $p_m = 1/(nn_{\text{row}})$,列变异概率 $p_m = 1/(nn_{\text{col}})$,即可变异对象中平均仅发生一次变异. 假定元素 $a_{ij} \in S_1$ 被选为变异对象,则有 $a'_{ij} = a_{ij} + rand(-\sigma_{ij}, \sigma_{ij})$;如果 $a'_{ij} < 1$,置 $a'_{ij} = rand(1,n)$;如果 $a'_{ij} > n^2$,置 $a'_{ij} = n^2 - rand(0,n)$. 设 $a_{kl} = a'_{ij}$,则置 $a'_{kl} = a_{ij}$,即将 $\boldsymbol{M}$ 中两元素 a_{ij} 与 a_{kl} 互换. $\sigma'_{ij} = \sigma_{ij} + rand(-1,1)$;如果 $\sigma'_{ij} < 1$ 或 $\sigma'_{ij} > \sigma_t$,则置 $\sigma'_{ij} = rand(1,\sigma_t)$.

因为变异概率与变异元素范围均由 n_{row} 与 n_{col} 决定,因此,元素置换算子是自适应变异算子.

行列置换算子　行置换与列置换算子,仅作用于半幻方,但保持半幻方的性质不变.

当 $n_{\text{row}} + n_{\text{col}} = 0, d_1 + d_2 \geqslant 1$ 时,随机取两行元素互换,或随机取两列元素互换.

半幻方演化过程中,元素置换算子被激活,但行列置换算子不工作. 当 $n_{\text{row}} + n_{\text{col}} = 0$ 时,进入对角幻和演化阶段,行列置换算子被激活,而元素置换算子失效.

3.3　启发式局部调整操作

演化算法进入稳定状态后,利用问题的启发式知识进行局部调整操作,可以提高演化算法的局部搜索效率. 通过行遍历与列遍历的搜索过程,如果两行或两列中一对以上元素互换之后可同时实现幻和,则可实行局部调整.

3.3.1　行列局部调整

逐行逐列比较,其中,$1 \leqslant k \leqslant n, 1 \leqslant l \leqslant n, 1 \leqslant s \leqslant n, 1 \leqslant t \leqslant n, k \neq l, s \neq t$. 行列局部调整操作仅作用于半幻方演化的后期,有利于半幻方的后期快速演化.

1) 互换一对元素的条件. 如果

$$\sum_{i=1}^{n} a_{ki} - c = c - \sum_{j=1}^{n} a_{lj} = a_{ks} - a_{ls},$$

则互换第 k 行与第 l 行中对应 s 列的两元素;如果

$$\sum_{i=1}^{n} a_{ik} - c = c - \sum_{j=1}^{n} a_{jl} = a_{sk} - a_{sl},$$

则可互换第 k 列与第 l 列中对应 s 行的两元素.

2) 互换二对元素的条件. 如果

$$\sum_{i=1}^{n} a_{ki} - c = c - \sum_{j=1}^{n} a_{lj} = a_{ks} + a_{kt} - a_{ls} - a_{lt},$$

则互换第 k 行与第 l 行中对应 s 列与 t 列的两元素;同样,如果

$$\sum_{i=1}^{n} a_{ik} - c = c - \sum_{j=1}^{n} a_{jl} = a_{sk} + a_{tk} - a_{sl} - a_{tl},$$

则可互换第 k 列与第 l 列中对应 s 行与 t 行的两元素.

行列局部调整元素对可增加至三对以上,因为三对元素以上的可调整条件的搜索代价过高,因此,仅搜索最基本的一对与两对元素的可互换条件.

3.3.2　对角局部调整

逐行或逐列比较,其中,$1 \leqslant i \leqslant n, 1 \leqslant j \leqslant n, i \neq j$;对角局部调整操作仅作用于对角幻和演化的后期,有利于对角幻和的后期快速演化.

如果 $a_{ii} + a_{ij} = a_{ji} + a_{jj}$,且

$$(a_{ii} + a_{jj}) - (a_{ij} + a_{ji}) = \sum_{k=1}^{n} a_{kk} - c,$$

则将 a_{ii} 与 a_{ji} 互换,a_{ij} 与 a_{jj} 互换.

如果 $a_{ij} + a_{i,n-i+1} = a_{n-j+1,j} + a_{n-j+1,n-i+1}$,

且

$$(a_{i,n-i+1} + a_{n-j+1,j}) - (a_{ij} + a_{n-j+1,n-i+1}) = \sum_{k=1}^{n} a_{n-k+1,k} - c,$$

则将 a_{ij} 与 $a_{n-j+1,j}$ 互换,$a_{i,n-i+1}$ 与 $a_{n-j+1,n-i+1}$ 互换.

如果 $(a_{ii} + a_{jj}) - (a_{ij} + a_{ji}) = \sum_{k=1}^{n} a_{kk} - c$,

且

$$(a_{i,n-i+1} + a_{j,n-j+1}) - (a_{i,n-j+1} + a_{j,n-i+1}) = \sum_{k=1}^{n} a_{n-k+1,k} - c,$$

则将 i 行与 j 行互换.

如果 $(a_{ii} + a_{jj}) - (a_{ij} + a_{ji}) = \sum_{k=1}^{n} a_{kk} - c$,

且

$$(a_{n-i+1,i} + a_{n-j+1,j}) - (a_{n-j+1,i} + a_{n-i+1,j}) = \sum_{k=1}^{n} a_{n-k+1,k} - c,$$

则将 i 列与 j 列互换.

如果

$$\begin{aligned}&(a_{ii} + a_{n-i+1,n-i+1}) - (a_{i,n-i+1} + a_{n-i+1,i})\\ &= \sum_{k=1}^{n} a_{kk} - c\\ &= c - \sum_{k=1}^{n} a_{n-k+1,k},\end{aligned}$$

则将 i 行与 $(n-i+1)$ 行互换.

3.4 算法

幻方演化算法中不使用杂交或重组算子，种群亲本数目的多少不影响算法的效率. 因此，算法仅用一个亲本，每代生成 10 个子代个体. 最小化问题的进化算法中目标函数与适应值成反比，即目标函数越小，个体的适应值越大. 在多亲本且具有重组算子的进化算法中，一般先将最小化问题转化成最大化问题，使适应值与个体的优劣性一致. 但在只有一个亲本的幻方演化算法中，可以直接采用其目标函数作为“适应值”. 在半幻方演化过程中，当最优个体的目标函数值较大时，亲本更新采用$(\mu,\lambda)-ES$机制；当最优个体的目标函数值小于 $50n$ 时，亲本更新采用$(\mu+\lambda)-ES$机制. 同样，在对角幻和演化过程中，当最优个体的目标函数值较大时，亲本更新采用 $(\mu,\lambda)-ES$ 机制；当最优个体的目标函数值小于 100 时，亲本更新采用$(\mu+\lambda)-ES$机制. 此外，初始变异方差取 $\sigma_{ij}=n^2, 1\leqslant i\leqslant n, 1\leqslant j\leqslant n$. 算法流程如图 3 所示.

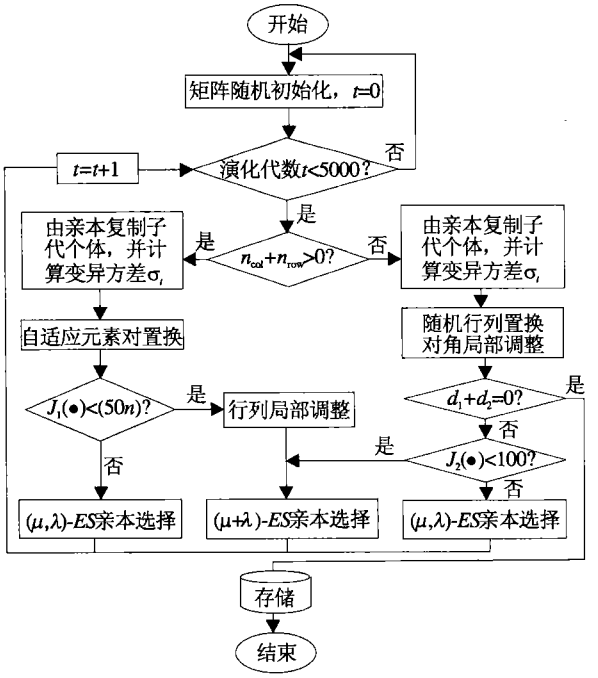

图 3 随机幻方构造自适应演化算法流程图

Fig. 3 Flowchart for the evolutionary algorithm of random magic squares

算法基本步骤

step1 初始化亲本，在矩阵中随机排列从 1 开始的 n^2 个最小自然数.

step2 （计算 σ_t 后）当 $n_{row}+n_{col}>0$ 时，采用适应值函数 $J_1(\boldsymbol{M})$，等概率选择 3 种元素置换算子之一，以相应变异概率对矩阵进行操作，而且当最优个体的适应值小于 $50n$ 时，还必须启动行列局部调整操作，生成下代个体. 当 $n_{row}+n_{col}=0$ 时，适应值函数改为 $J_2(\boldsymbol{M})$，采用行列置换算子与对角局部调整操作，生成下代个体.

step3 亲本更新，根据不同情况从当前种群 Ω_t 中或子代 Θ_t 选择下代亲本.

当 $n_{row}+n_{col}>0$ 时，如果当前最优个体的适应值 $J_1(\boldsymbol{M})\geqslant 50n$，则采用 $(\mu,\lambda)-ES$ 更新机制；否则，采用 $(\mu+\lambda)-ES$ 更新机制.

当 $n_{row}+n_{col}=0, d_1+d_2\geqslant 1$ 时，如果当前最优个体的适应值 $J_2(\boldsymbol{M})\geqslant 100$，则采用 $(\mu,\lambda)-ES$ 更新机制；否则，采用 $(\mu+\lambda)-ES$ 更新机制.

step4 如果在第 2 阶段最优个体的适应值 $J_2(\boldsymbol{M})=0$，即 $d_1+d_2=0$，随机构造的矩阵满足幻方条件，算法终止；否则，转 step2 继续.

3.5 计算复杂度分析

进化计算的计算复杂度主要由适应值函数的复杂度决定. n 阶幻方演化算法的适应值函数的计算复杂度为 $(2n+2)n$，该算法一个个体每代最多可能经历 4 种变换形式，因此，每代一个个体必须计算 4 种可能变换形式的适应值. 但在最坏情况下，该算法的计算复杂度主要由局部调整操作决定. 每代演化过程中各种操作在最坏情况下的计算复杂度可以分别给出. 半幻方演化过程中，自适应变异操作与行列局部调整操作的计算复杂度分别为

$$\frac{2}{3}n(n_{col}+n_{row})+\frac{1}{3}n^2$$

与

$$(n^2+n)n((n-n_{col})^2+(n-n_{row})^2).$$

对角幻和演化过程中，行列置换操作与对角局部调整操作的计算复杂度分别为 $2n$ 与 $5n^2$. 行列局部调整操作与对角局部调整操作分别仅作用于两阶段演化过程的后期.

4 随机幻方演化分析

数值分析的目的是研究幻方阶数与算法计算时间之间的关系. 从 10 阶幻方开始，阶次依次递增一倍，直至 100 阶幻方，每阶幻方计算 10 次，每次计算取不同的随机化初始亲本. 设每阶幻方搜索的成功次数为 n_{suc}，取每阶幻方的平均计算时间为算法的计算时间代价. 算法计算时间分为两部分，第 1 部分为半幻方的演化时间 t_{rc}，算法演化代数 g_{rc}；第 2 部分为在半幻方基础上进行对角幻和演化的计算时间

表 1　平均演化时间、平均演化代数 vs. 幻方阶数

Table 1　Average evolution time and generations vs. the order of magic squares

	10	20	30	40	50	60	70	80	90	100
t_n	0.8	12	84	253	641	1197	1959	3226	4817	9140
t_d	0.2	2	10	42	115	313	341	575	1028	1328
g_n	220	885	3432	4710	8630	10028	13046	16152	18041	30744
g_d	319	492	2475	1861	2188	4897	3777	3765	5800	3364
n_{suc}	10	10	10	10	10	10	10	10	10	10

t_d,算法演化代数 g_d. 计算结果如表 1 所示,横标数据为幻方的阶数. 计算机处理器采用 PentiumIII,主频为 450MHZ,内存 64M.

从表 1 计算结果可知,每阶幻方连续 10 次不同搜索都是成功的,而且每次均得到不同的随机幻方. 对角幻和的演化时间在幻方演化全过程中所占比例是较大的,且基本上保持一定比例. 如果按演化代数,对角幻和的演化代数所占比例更大,但随幻方阶数逐步下降. 图 4 与图 5 分别是幻方演化算法的目标函数值和满足幻和的行列和对角线数目与演化代数的关系,图中自左至右曲线对应幻方阶数分别为:40,50,60,70,80,90,100. 图 4 中每阶幻方离散性能曲线的末端稳定部分出现短暂振荡,是半幻方演化成功并向对角幻和演化过度时更换目标函数所致. 可以看出,半幻方的演化过程可分为前期、中期与后期. 演化前期主要缩小每行每列与幻和的差值,虽然目标函数值急剧下降,但满足幻和的行列数基本为零;演化中期,行列逐步向幻和逼近,目标函数值下降趋缓,但满足幻和的行列数振荡上升;演化后期,行列快速趋近幻和,目标函数值下降缓慢,但满足幻和的行列数急剧上升. 半幻方演化的 3 个阶段反映了元素置换算子的自适应性质.

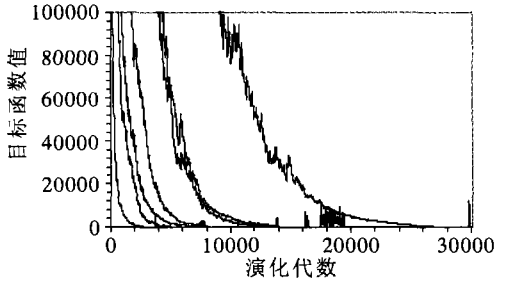

图 4　目标函数值与演化代数的对比

Fig. 4　Objective value vs. number of evolution generations

此外,如果在幻方的整个演化过程中采用统一的$(\mu+\lambda)-ES$ 亲本更新机制,演化算法很难收敛到幻方条件,算法容易陷入局部最优,即部分满足幻方条件. 如果采用简单演化算法中的统一目标函数,算法很难构造 10 阶以上幻方.

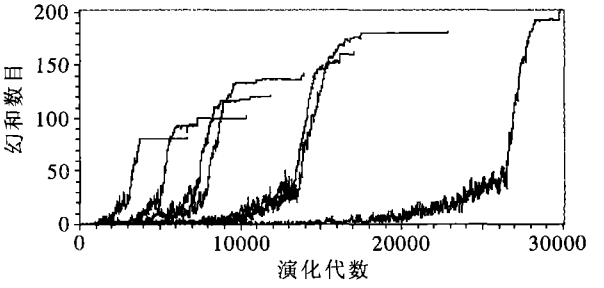

图 5　幻和数目与演化代数的对比

Fig. 5　Number of magic units vs. number of evolution generations

5　讨论与研究建议

虽然幻方分步构造猜想有待证明,但数值实验结果表明,建立在该猜想条件下的随机幻方演化算法表现出极限成功率,即每次幻方演化过程都能得到不同的随机幻方,而且演化算法具有较高的构造效率. 幻方演化算法属于随机构造法,不同于传统的确定式幻方构造方法. 因此,幻方演化算法是幻方构造方法中重要的随机自适应构造新方法.

幻方演化算法的高演化效率源于幻方的分步构造猜想与自适应的元素置换算子. 幻方分步构造猜想将一个幻方的构造过程分解为两步,即半幻方演化构造与对角幻和演化构造. 这种分步构造法将行列幻和的构造过程与对角幻和的构造过程"解耦",使之互不影响. 自适应的元素置换算子可将变异对象定位于未满足幻和的行列元素,并在半幻方构造过程中自适应调整变异概率,使平均变异元素个数保持一定.

幻方演化算法的极限成功率源于行列与对角局部调整操作. 在半幻方演化的后期,对于难以通过随机元素置换算子实现幻和构造的局部元素置换对,行列局部调整操作在启发式知识下,搜索这些满足

条件的元素对实现置换.在对角幻和的演化后期,对角局部调整操作具有同样效果.

演化算法不仅能构造一般的高阶幻方,而且也能构造具有附加条件的特殊幻方.2次幻方(doubly magic square)与3次幻方(trebly magic square)的构造是随机幻方演化算法研究中有待进一步探索的问题.

幻方数量的天文数字与随机幻方的演化构造算法具有密码学的基本特征,研究与发现基于幻方问题的新密码学加密原理是幻方演化构造算法最有前景的新研究方向[8-10].

参 考 文 献

[1] Berlekamp E R, Convay J H, Guy R K. Winning Ways for Your Mathematical Plays. London, UK: Academic Press, 1982

[2] Madachy L S. Magic and Antimagic Squares // Madachy J S, ed. Madachy's Mathematical Recreations. New York, USA: Dover, 1979: 85-113

[3] Pinn K, Wieczerkowski C. Number of Magic Squares from Parallel Tempering Monte Carlo. International Journal of Modern Physics, 1998, 9: 541-547

[4] Abe G. Unsolved Problems on Magic Squares. Discrete Mathematics, 1994, 127(1/2/3): 3-13

[5] Kraitchik M. Magic Squares // Kraitchik M, ed. Mathematical Recreations. New York, USA: Norton, 1942: 142-192

[6] Xie Tao, Kang Lishang. An Evolutionary Algorithm for Magic Squares // Proc of the Congress on Evolutionary Computation. Canberra, Australia, 2003, Ⅱ: 906-913

[7] Bäck T, Hoffmeister F, Schwefel H P. A Survey of Evolution Strategies // Proc of the 4th International Conference on Genetic Algorithms. San Diego, USA, 1991: 2-9