

双适应函数单亲遗传算法

范小勤¹, 胡能发²

(1. 广州番禺职业技术学院 基础课部, 广州 511483; 2. 韩山师范学院 数学与信息技术学院, 广东 潮州 521041)

(fan66412@163.com)

摘要:设计了一种新的双适应函数单亲遗传算法,该算法对个体和基因分别计算其适应值,并将适应值最差的基因进行变异,从而大大地提高了遗传算法的全局收敛速度。实验表明,该算法比只有一个适应值的传统遗传算法具有更快的收敛速度。

关键词:单亲遗传算法;基因换位;评估函数

中图分类号: TP273 **文献标志码:** A

Partheno-genetic algorithm based on dual fitness

FAN Xiao-qin¹, HU Neng-fa²

(1. Department of Basic Courses, Guangzhou Panyu Polytechnic, Guangzhou Guangdong 511483, China;

2. College of Mathematics and Information Technology, Hanshan Teachers College, Chaozhou Guangdong 521041, China)

Abstract: In this paper a new partheno-genetic algorithm was proposed. It calculates the fitness of individuals and genes respectively, and changes the worst gene. It has greatly improved the global convergence of genetic algorithm (GA). The simulation results show that the new algorithm not only can search the global optimum solution in a short time, but also has excellent stability.

Key words: partheno-genetic algorithm; gene exchange; evaluation function

0 引言

遗传算法是一种模拟生物进化过程中自然选择和自然遗传机制的随机全局优化搜索算法^[1]。近年来,遗传算法以其高效和实用的特点在各个领域得以广泛应用,如神经网络、函数优化、图像处理、系统辨识、专家系统等等。用遗传算法求解一个问题的一般步骤是:

- 1) 分析问题,确定一种编码方案;
- 2) 确定问题的目标函数,设计一个适应函数;
- 3) 生成一个含有若干个个体的初始群体,每个个体均是问题的一个可行解;
- 4) 计算群体中每个个体的适应值;
- 5) 如果找到问题的最优解,退出循环;
- 6) 否则,根据每个个体的适应值,进行选择、杂交、变异等遗传操作;
- 7) 返回 4)。

传统的遗传算法主要是通过母体杂交算子产生后代,当母体相同时,后代将停止进化或出现“早熟现象”,因此这种遗传算法往往要求群体具有足够的多样性。而单亲遗传算法^[2-4]是一种不使用母体杂交,而仅仅通过单个个体繁殖后代的一种特殊遗传算法。在现有的遗传算法中,包括单亲遗传算法,大多只有一个适应函数,用于指导算法搜索的方向,而遗传算法中的主要算子如变异等,则采用随机方式来选取,具有一定的盲目性,而这种盲目性最终导致了算法收敛速度的下降。基于这一思想,本文采用单个个体组成初始群体,在

个体评估的基础上,增加了一种基因评估函数,除了对个体进行评估以外,还对个体的每个基因进行评估。在单亲遗传算法的变异中,根据基因评估函数选择出最差的基因进行变异,这样,变异后的基因将以较大的概率优于原基因,所产生的后代将以较大的概率优于母体,从而避免了演化的盲目性,加快了求解速度。

1 单亲遗传算法

单亲遗传算法是针对组合优化问题提出来的,它采用序号编码方式,但不使用传统遗传算法的序号编码杂交算子,它仅在一条染色体上进行变异操作。单亲遗传算法的基因换位算子是按一定的概率 p_c 交换一条染色体中某些位置上的基因,被交换基因的位置是随机的。

用遗传算法求解组合优化问题时,一般宜采用序号编码,但采用序号编码的遗传算法不能使用普通的杂交算子,必须使用部分匹配交换、圈交换、序号交换等特殊的杂交算子,但这些杂交算子操作复杂,且效率不高。单亲遗传算法针对序号编码的上述不足,取消了序号编码的杂交算子,代之以仅在一条染色体上操作的基因换位等遗传算子。由于单亲遗传算法不使用杂交算子,即使种群中各个个体均相同,也不影响遗传迭代的进行,摆脱了对群体多样性的要求,简化了遗传操作,提高了搜索效率,而且不存在传统遗传算法常见的“早熟收敛”问题。单亲遗传算法中的基因换位可分为单点换位和多点换位。单点换位是一次只交换一对(两个)基因的位置;多点换位是对预先给定的正整数 μ ,取随机数 $i(1 \leq i \leq \mu)$,

收稿日期:2009-01-07;修回日期:2009-03-19。 基金项目:广东省教育科研项目(07JT130)。

作者简介:范小勤(1966-),男,湖北黄冈人,副教授,硕士,主要研究方向:系统最优化、遗传算法、综合评价; 胡能发(1962-),男,湖北荆州人,副教授,硕士,CCF 会员,主要研究方向:演化计算、信息安全、软件工程。

然后交换 i 对位置的基因值。如图1和图2所示。

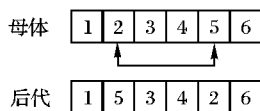


图1 单点基因换位

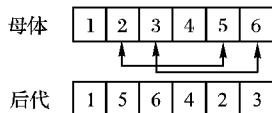


图2 多点基因换位

2 双适应函数单亲遗传算法的基本思想

根据孟德尔的遗传学原理,如果生物体基因发生变异,则对应生物体相关部位的性质就会发生相应的改变。如果在生物进化过程中尽量保持好的基因不变,而对不良基因加以改造,使之发生突变,这将有助于提高生物体的适应率。基于这一思想,在单亲遗传算法中,我们不仅对个体的适应值进行评估,而且对个体中的每个基因的适应值也进行评估,然后根据评估的结果找出个体中的最差基因,并对其进行变异操作,从而以较高的概率避免了好的基因向不好的基因发生变化的情况发生。为了保证个体不至于陷入局部全优,在单亲遗传算法的变异算子中,还以相对较少的概率接受较差的个体和基因。

以下以 n 皇后问题^[5-6]为测试实例,来阐述这一思想。

n 皇后问题中,要求所有皇后不在同一行、同一列和同一斜线上,可设一个可行解为 $P = \{a[1], a[2], \dots, a[n]\}$, 这里 $a[i] \in \{1, 2, \dots, n\}$, $i = 1, 2, \dots, n$ 表示第 i 行第 $j = a[i]$ 列放有一个皇后,故 n 皇后问题应满足的约束条件为:

- 1) 任意两个皇后不在同一列, 即 $a[i] \neq a[j]$ (当 $i \neq j$ 时);
- 2) 任意两个皇后不在同一斜线上, 即 $|a[i] - a[j]| \neq |i - j|$ (当 $i \neq j$ 时);

设 n 皇后问题的一个布局(即个体)为 $P = \{a[1], a[2], \dots, a[n]\}$, 定义:

$$Appetence_{ij} = \begin{cases} 1, & |a[i] - a[j]| \neq |i - j| \\ 0, & \text{其他} \end{cases}$$

称 $Appetence_{ij}$ 为基因 i 与基因 j 的亲密度。

从亲和度的定义可知,当两个基因不在同一对角线上时,亲和度为1,表明其基因是优质基因,否则亲和度为0,其基因劣质基因。

$$\text{定义 } fAppetence_i = \sum_{j=1}^n Appetence_{ij}$$

称 $fAppetence_i$ 为个体第 i 个基因的适应值。

从基因的适应值定义可知,个体的第 i 个基因的适应值即为基因 i 与其他所有基因亲和度的总和。在实施遗传算子的变异操作中,将对适应值较小的基因进行变异。

$$\text{定义 } f = \sum_{i=1}^n fAppetence_i$$

称 f 为个体的适应函数,个体的适应函数值即为各基因适应值的总和。

由以上定义可知,当 $fAppetence_i$ 越大时,说明基因越好,否则越差;而当 f 越大时,说明个体越好。在演化过程中,个体的适应值和基因的适应值都会越来越大。在用单亲遗传算法求解问题时,个体的适应值用于指导后代的选择,而基因的适应值则用于指导选择个体中对哪些基因进行变异。当所有

的基因适应值 $fAppetence_i = n - 1$ 时, $f = n \times (n - 1)$, 此时的个体,即为 n 皇后问题的解。

3 双适应函数单亲遗传算法

算法初始群体中只含有一个个体,将该个体进行遗传操作后所得的新个体如果优于母体,则用新个体代替母体作为下一代演化的母体,否则,下一代演化的母体不变。为了防止算法陷入局部最优,算法中还以一定的概率 p (本文 p 的值为0.01)接受较差的新个体。具体算法如下(其中假定随机函数 $\text{rand}()$ 产生0到32767之间的任意整数,随机函数 $\text{rnd}()$ 产生0到1之间的任意小数)。

输入:皇后个数 n , 概率 p

输出:皇后问题的一个解,即数组 $a[n]$

步骤1 随机生成只包含一个个体的初始群体即数组 $a[n]$, 当 $i \neq j$ 时, $a[i] \neq a[j]$, 其中 $i, a[i] \in \{1, 2, \dots, n\}$, 具体处理过程如下代码所示。这样处理的目的是保证了初始群体满足任意两个皇后不在同一列的限制,而且保证了其随机性。

```
for (i = 1; i <= n; i++)
    b[i] = i;
for (i = 1; i <= n; i++) {
    j = 1 + rand() % (n - i + 1);
    a[i] = b[j];
    b[j] = b[n - i + 1];
}
```

步骤2 计算每个个体 $a[i]$ 的亲密度 $Appetence[i, j]$, 基因的适应值 $fAppetence[i]$, 个体的适应值 f ;

步骤3 演化。

```
while (fnew != n * (n - 1)) {
    求出最差基因的位置 e1;
    e1 = argmin_{i \in \{1, \dots, n\}} fAppetence[i];
    随机产生另一个基因的位置 e2;
    e2 = 1 + rand() % n;
    单点换位操作:
    即交换 e1, e2 两个位置上的基因值得到后代新个体;
    计算新个体的亲密度 AppetenceNew[i, j], 基因的适应值 fAppetenceNew[i], 新个体的适应值 f_new, 其中 i, j = 1, 2, \dots, n, i \neq j;
    if (f_new > f) or (rand() < p)
        新个体代替旧个体;
}
```

步骤4 输出演化结果 $a[n]$ 。

4 实验分析

为了验证算法的有效性,本文对不同数量的皇后问题在不同配置的计算机上进行了实验验证。

实验1 当 $n = 100$ 时,在 P4 1.5 GHz CPU, 256 MB 内存的计算机上运行得到如下皇后问题实验数据:

84 39 79 27 16 31 5 70 91 83 50 22 10 21 64 81 63 36 77
52 15 6 92 62 49 43 33 40 48 98 34 66 94 42 28 19 93 65 44 2
85 82 96 13 67 76 11 100 4 95 41 32 88 53 68 17 61 74 3 35 26
47 1 90 56 80 38 7 89 99 37 9 46 14 75 97 86 24 58 30 8 71 60
20 87 45 55 23 12 57 73 78 69 72 25 59 54 51 18 29

实验2 当 $n = 500$ 时,在同一台计算机上运行得到如下皇后问题实验数据:

454 236 101 318 258 367 381 168 206 345 107 233 434 416

458 143 330 105 214 494 169 208 189 492 153 121 221 373 414
 220 171 319 209 155 361 455 229 371 369 484 98 247 230 267
 443 469 305 66 344 413 200 263 144 405 91 165 271 5 299 26
 467 148 224 386 223 250 390 365 80 173 269 425 332 194 85
 210 280 428 160 297 195 135 379 394 473 205 67 113 359 463
 402 372 110 70 166 60 116 162 246 231 136 33 31 277 43 315
 272 150 57 309 289 415 301 2 4 471 36 19 438 74 199 498 216
 96 59 252 178 8 245 307 61 102 460 241 182 176 384 417 35
 196 115 253 483 279 477 53 185 286 447 211 29 44 292 306 40
 364 242 314 391 186 274 65 331 77 396 203 103 273 108 445
 183 429 406 380 6 497 322 407 177 295 472 111 450 397 235
 304 453 481 288 239 418 276 448 129 90 180 95 500 366 163
 325 62 68 464 248 78 22 491 339 466 362 174 181 38 64 122
 490 442 12 52 23 175 408 237 93 430 139 489 119 222 311 219
 100 156 480 63 46 81 432 329 94 352 161 377 347 437 499 327
 190 316 55 179 21 337 323 112 446 346 388 476 228 217 256
 207 487 215 449 486 120 427 341 37 333 79 308 275 451 89 356
 1 488 423 17 45 468 28 404 114 421 296 441 106 147 7 399 131
 298 284 435 202 470 152 34 358 360 291 461 382 71 16 482 374
 99 191 474 25 479 9 11 495 146 342 69 240 58 172 86 157 334
 376 475 75 32 82 313 293 262 439 20 227 478 30 395 125 410
 354 170 393 310 496 324 39 117 56 137 15 336 27 3 10 151 389
 357 431 84 266 411 392 132 140 124 212 403 370 159 184 109
 54 234 419 193 440 465 353 51 123 126 436 300 48 485 278 493
 422 232 50 343 142 243 149 188 127 282 387 133 164 351 76
 302 226 328 14 192 138 130 348 264 420 378 238 118 158 251
 401 225 197 257 290 433 141 452 24 398 244 368 72 285 134
 462 400 456 88 104 338 349 249 198 87 312 83 409 92 13 201
 326 459 270 49 97 457 218 261 73 281 145 260 268 320 444 340
 47 128 294 383 213 335 254 204 363 42 426 265 154 303 18 355
 321 385 350 259 283 412 187 167 255 41 317 424 287 375

从运行的结果看,尽管遗传算法理论上的结果是近似解,

但本文算法所得的结论却是精确解,这充分反映了本算法的优点。当 $n = 100$ 时,耗时约 124 ms; 当 $n = 500$ 时,耗时约 19 s,并且多次在 1 s 以内找到问题的解,实验中没有发现不收敛的情况。另外,求解 1000 皇后问题所用时间平均只有约 3 min,其中多次在 1 min 以内找到该问题的解,而求解 3000 个皇后所用时间平均约为 55 min,有时能在几分钟内就能得到该问题的精确解。而对于没有基因评估函数的算法,其求解速度几乎要慢 10 倍以上,而且随着皇后数量的增加,其收敛速度的差别会更大。

5 结语

采用个体评估与基因评估双适应函数,将最差的基因进行变异,改善了单亲遗传算法搜索全局最优解的方向,避免了算法对个体基因的盲目变异,从而加快了演化速度。实验结果表明,该算法具有演化速度快、全局性能好、不易陷入局部最优解的特点。当然,由于该算法只对一个个体进行演化,初始群体的好坏对演化速度有一定的影响。另外 n 皇后问题较易设计出其基因评估函数,具有一定的特殊性,能否将个体评估和基因评估双重评估方法推广到一般的情形,使其能解决一般的优化问题,有待进一步研究。

参考文献:

- [1] 王小平,曹立明. 遗传算法—理论、应用与软件实现[M]. 西安: 西安交通大学出版社, 2002.
 - [2] 杜健,李波. 对于一维原材料切割问题的单亲遗传启发方法[J]. 计算机工程, 2006, 32(2): 234 - 236.
 - [3] 祝延军,胡纯德,高随祥. 最短路问题的改进单亲进化遗传法[J]. 计算机工程与应用, 2005, 41(8): 64 - 66.
 - [4] 张楷波,祝延军. 基于 PEGA 求解 TSPD 的物流配送路径优化算法[J]. 计算机工程与设计, 2006, 27(12): 2270 - 2272.
 - [5] 周康,同小军,许进. 基于闭环 DNA 模型的八皇后问题算法[J]. 计算机工程与应用, 2007, 43(6): 4 - 6.
 - [6] 邓宏涛,朱珣. N 皇后问题 Las Vegas 优化算法的实现[J]. 江汉大学学报: 自然科学版, 2006, 34(4): 56 - 58.
- (上接第 1876 页)
- 从图 6 和 7 可以看出,在子节点无负载时,计算出预测块大小比随机指定块大小性能平均提高 1.2%,在子节点有负载时,计算出预测块大小比随机指定块大小性能平均提高 6.4%,因此无论是有负载还是没有负载,在数据和计算均密集型的应用中,实时计算出来预测部分大小所取得的性能比随机指定的性能要好。
- ## 4 结语
- 本文针对现有网格环境中预测方法存在的问题,提出了基于分块的预测的方法。该方法的主要特征是在预测时利用数据块的前一部分的执行时间来预测剩下部分的时间,达到实时、精确的效果。实验证明,该方法与现有预测方法相比,具有更好的执行效率和通用性。
- ## 参考文献:
- [1] YANG CHAO-TUNG, CHENG KUAN-WEI, SHIH WEN-CHUNG. On development of an efficient parallel loop self-scheduling for grid computing environments[J]. Parallel Computing, 2007, 33(7/8): 467 - 487.
 - [2] MAHESWARAN M, ALI S, SIEGEL H J, et al. Dynamic matching and scheduling of a class of independent tasks onto heterogeneous computing systems[C]// Proceedings of the 8th IEEE Heterogeneous Computing Workshop. Washington, DC: IEEE Computer Society, 1999: 30 - 44.
 - [3] PHINIJAROENPHAN P, BEVINAKOPPA S, ZEEPHONGSEKUL P. A method for estimation the execution time of a parallel task on a grid node[C]// EGC 2005, LNCS 3470. Berlin: Springer, 2005: 226 - 236.
 - [4] IVERSON M, OZGUNER F, POTTER L. Statistical prediction of task execution times through analytic benchmarking for scheduling in a heterogeneous environment[C]// Proceedings of the 8th Heterogeneous Computing Workshop. Washington, DC: IEEE Computer Society, 1999: 15 - 29.
 - [5] 栾翠菊,宋广华,郑耀,等. 一种网格并行任务执行时间预测算法[J]. 计算机集成制造系统, 2007, 13(9): 1806 - 1810.
 - [6] 韩耀罗,罗雪梅. 网格计算环境下任务执行时间的组合预测[J]. 计算机工程, 2006, 32(21): 70 - 72.