American Journal of Computational Mathematics, 2016, 6, 32-36
Published Online March 2016 in SciRes. http://www.scirp.org/journal/ajcm
http://dx.doi.org/10.4236/ajcm.2016.61004

An Integer Coding Based Optimization Model for Queen Problems

Nengfa Hu

Department of Computer Science and Engineering, Hanshan Normal University, Chaozhou, China
Email: hunengfa326@sina.com

Received 4 March 2016; accepted 27 March 2016; published 30 March 2016

Abstract

Queen problems are unstructured problems, whose solution scheme can be applied in the actual job scheduling. As for the n-queen problem, backtracking algorithm is considered as an effective approach when the value of n is small. However, in case the value of n is large, the phenomenon of combination explosion is expected to occur. In order to solve the aforementioned problem, queen problems are firstly converted into the problem of function optimization with constraints, and then the corresponding mathematical model is established. Afterwards, the n-queen problem is solved by constructing the genetic operators and adaption functions using the integer coding based on the population search technology of the evolutionary computation. The experimental results demonstrate that the proposed algorithm is endowed with rapid calculation speed and high efficiency, and the model presents simple structure and is readily implemented.

Keywords

Queen Problem, Function Optimization, Mathematical Model, Evolutionary Computation, Integer Coding

1. Introduction

As an old and well-known problem, the eight queens problem proposed by the international chess player, Marx Bethel, in 1848 is a typical case of the backtracking algorithm [1]. Since then, it has been studied by many mathematicians including Gauss and Cantos, and is generalized as the layout problem of the n-queen. The n-queen problem is an unstructured problem, which can be effectively solved using approaches including the search strategy based on artificial intelligence and the backtracking algorithm [2]-[5]. Some components, such as multi-leveled equations, graphics, and tables are not prescribed, although the various table text styles are provided. The formatter will need to create these components, incorporating the applicable criteria that follow.

How to cite this paper: Hu, N.F. (2016) An Integer Coding Based Optimization Model for Queen Problems. *American Journal of Computational Mathematics*, **6**, 32-36. http://dx.doi.org/10.4236/ajcm.2016.61004

As to the definition of the n-queen problem: n queens are placed on the international checkerboard with the dimension of the grid being n × n, and the precondition is that no queen suffers from the attack from another queen. That is to say, among the n queens, any two of them are not on the same row, column, or oblique line. There are many algorithms to solve this problem, such as the exhaustive method, backtracking algorithm, learning algorithm of the neural network, and so on. However, when the value of n is great, these algorithms take a long time for computation and therefore become infeasible.

Evolutionary computation is a global optimization based random search algorithm that simulates the natural selection and natural genetic mechanism of the biological evolution. Evolutionary computation has been applied in various fields including neural network, function optimization, image processing, system identification, expert system and so on due to its high efficiency and practicality through years of study. As for the application in solving the n-queen problems, evolutionary computation is an efficient nondeterministic algorithm [6]-[10].

2. Modeling and Evolutionary Algorithm for n-Queen Problems

The general steps for solving a problem using the evolutionary computation are demonstrated as follows:

1) Analyzing the problem to determine a coding scheme;

2) Determining the objective function of the problem, to design a fitness function;

3) Generating an initial population that contains several individuals, each of which is a feasible solution of the problem;

4) Calculating the fitness values of each individual in the population;

5) If the optimal solution of the problem is found, quitting the cycle;

6) Performing the genetic operations including selection, hybridization and variation according to the fitness values of each individual;

7) Returning to step (4).

It is regulated in the n-queen problems that all queens are not in the same row, column, and oblique line. Assume that a layout of the n-queen problems is $P=\{a[0],a[1],\cdots,a[n-1]\}$, where, $a[i]\in\{0,1,\cdots,n-1\}$ represents that a queen is placed on the i^{th} row and $a[i]^{th}$ column. Then, the n-adic queen problems are supposed to satisfy the following constraints:

1) Any two queens are not in the same column, that is $a[i]\neq a[j]$, (when $i\neq j$).

2) Any two queens are not located in the same oblique line, namely, $|a[i]-a[j]|\neq|i-j|$, (when $i\neq j$).

A random function can be utilized to process the constraint 1, and the specific processing procedure is demonstrated as follows (**Figure 1**):

Constraint 2 can be converted into the objective function *f*:

Suppose that $a[0],a[1],\cdots,a[n-1]$ is an individual.

Let $f=\sum_{i=0}^{n-2}\sum_{j=i+1}^{n-1} f_{ij}$, where $f_{ij}=\begin{cases}1 & \text{when } |a[i]-a[j]|\neq|i-j| \\ 0 & \text{otherwise}\end{cases}$

According to the definition of *f*, when $f=(n-1)+(n-2)+\cdots+1=n(n-1)/2$, we obtain $|a[i]-a[j]|\neq|i-j|$. Thereby, it is obtained that the individuals under such circumstance satisfy the requirements of the constraint 2.

```
for i: =0 to n-1 do
    b[i]: = i
for i: = 0 to n-1 do
BEGIN
    j: = random(n-i)
    a[i]: = b[j]
    b[j]: = b[n-i-1]
END
```

Figure 1. Condition 1 code.

Obviously, the greater the function value is, the smaller the possibility of the mutual attacks between queens. When the function value f reaches the maximum $n(n-1)/2$, the individual corresponding to the maximum value is the solution of the problem. Therefore, the function value f can be defined as the fitness function of the individuals.

3. Algorithm

1) The aforementioned program segments are utilized to generate the initial population $\{(p[0,0],\ p[0,1],\cdots,\ p[0,n-1]),(p[1,0],\ p[1,1],\cdots,\ p[1,n-1]),\cdots,(p[NN-1,0],\ p[NN-1,1],\ \ldots,\ p[NN-1,n-1])\}$, where, NN and n represent the population size and the number of the queens respectively. The first and second subscripts in array p indicate the serial numbers of the individuals and genes separately, and are called the individual and gene respectively for the sake of convenience;

2) The fitness values $f[i],(i=0,1,\cdots,NN-1)$ of all the individuals in the initial population are calculated, and the individual k_{max} with the maximum value being f_{max} and k_{min} with the minimum value being f_{min} are found out;

3) Hybridization: the location based crossover operator is adopted, and the specific process is demonstrated as follows: to begin with, a random integer c with the value varying from 0 to n is randomly generated to be regarded as the crossover point. Afterwards, the optimal individual kmax is selected from the population, followed by the generation of a random integer m (m $\neq k_{max}$ and m $\neq k_{min}$) with the value in the range of 0~NN. Then, the location based hybridization process is performed between the individual corresponding to the integer m and the optimal individual kmax at the point c. If the preferable individual in the hybrid progenies is superior to the optimal individual kmax, the worst individual kmin is replaced by the individual; otherwise, the worst individual kmin is reserved. The crossover operator is demonstrated in **Figure 2**, and its implementation algorithm is illustrated in **Figure 3**.

As demonstrated in **Figure 3**, the array element p[*,i] represents the value of the *i*th gene of the individual *, while ch1[i] and ch2[i] indicate the values of the *i*th gene in two generations respectively.

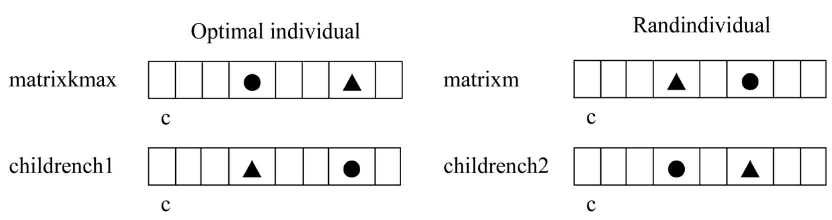

Figure 2. The crossover operator.

```
for i:=0 to n-1 do
begin
   ch1[i]:=p[kmax,i];
ch2[i]:=p[m,i];
end;
ch1[c]:=p[m,c];
ch2[c]:=p[kmax,c];
for i:=0 to n-1 do
if (p[kmax,i]=p[m,c]) and(i<>c) then
         ch1[i]:=p[kmax,c];
for i:=0 to n-1 do
if (p[m,i]=p[kmax,c]) and(i<>c) then
ch2[i]:=p[m,c];
```

Figure 3. The crossover algorithm.

4) Variation: the locations of any two genes in the original optimal individual k_{max} in step 3 are exchanged, to generate a new individual. If the new individual is superior to the previous optimal individual k_{max}, the latter is replaced by the former; otherwise, the latter is reserved.

5) Recalculating the values of fitness, maximum and minimum of the individuals changed by the hybridization and variation processes;

6) The results are output if the termination condition is satisfied, that is, output all the individuals i that satisfy the condition; otherwise, step 3 is repeated.

4. Operation Results

By using the proposed algorithm nearly all the solutions of the 4 queens problem (n = 4) are obtained in the test. Limited by the length of the paper, merely parts of the operation results are demonstrated, as illustrated in **Table 1**.

The value of the population size in the algorithm is generally taken as 10 - 20 times of the number of the queens. The evolution generation can be adjusted in the range of 50 - 100 times of the number of the queens, and the termination condition of the evolution is $f_{max} = n(n-1)/2$.

5. Conclusion and Improvement

The construction of the fitness functions using integers shortens the evaluation time of the computers, and thus greatly improves the calculation efficiency. Meanwhile, integer coding is performed on the individuals, which applies the knowledge of the n-queen problems and makes the algorithm concise and visual. The experimental results demonstrate that this algorithm is endowed with rapid calculation and high efficiency. Of course, in order to further improve the operating speed, several crossover points can be added according to the variation of the

Table 1. Experimental results.

Queens	Partial results
8	61528374 73025164 48157263 35714286 52617483 25703641 47302516 36824175
100	78 63 53 96 89 20 22 93 73 83 62 8 55 74 43 79 64 37 42 26 21 2 52 29 7 88 76 100 69 97 33 4 17 58 6 30 46 95 57 47 77 12 81 23 61 24 38 91 10 54 86 50 66 9 87 18 56 59 70 80 11 35 48 98 90 99 19 85 72 1 13 36 5 60 41 14 67 34 44 40 49 65 27 94 82 75 45 3 32 15 68 51 16 31 71 84 25 28 39 92
200	151 49 5 162 112 48 28 69 75 130 34 199 143 196 50 72 165 152 86 180 105 115 121 82 167 198 114 147 139 175 56 100 67 183 138 122 200 190 52 36 106 104 155 45 119 46 185 102 76 64 29 174 110 117 84 22 79 17 154 4 40 93 96 163 77 27 181 113 146 26 172 55 92 191 177 131 39 65 42 30 150 51 187 88 13 186 192 35 133 19 37 10 182 24 83 188 137 168 20 57 144 95 9 31 78 189 156 68 194 21 43 1 184 15 7 142 145 85 169 118 157 12 126 170 63 107 91 136 38 8 73 108 18 116 124 153 32 179 70 53 61 193 47 16 161 123 54 178 160 44 109 158 89 87 80 176 148 3 98 2 58 74 195 25 149 111 120 6 159 141 90 171 66 128 140 11 94 129 103 71 166 197 33 99 101 127 60 41 132 164 125 62 173 97 135 59 134 14 81 23
500	468 148 209 345 55 142 240 130 448 109 316 151 303 137 255 304 282 24 107 446 492 373 187 4 169 220 119 389 15 190 171 140 1 360 265 352 105 275 243 372 42 230 431 474 387 197 161 52 274 227 39 41 132 264 367 330 351 97 250 194 462 348 176 18 320 44 7 121 232 292 158 239 407 216 201 325 297 99 183 93 374 32 178 409 450 164 5 396 208 246 66 291 54 491 334 177 287 436 406 300 312 139 327 301 113 390 112 155 376 388 185 217 26 333 144 391 17 363 188 251 484 36 354 467 286 302 238 47 485 184 480 463 231 430 57 206 379 199 103 479 114 3 440 405 314 223 453 439 296 76 136 9 172 28 399 219 470 486 261 157 50 321 281 267 464 259 125 357 397 418 253 27 195 456 433 328 393 116 81 465 98 435 442 51 10 478 263 35 48 34 306 38 211 444 404 84 473 83 122 289 67 384 82 237 248 43 364 6 86 94 381 258 180 426 340 451 8 70 283 454 257 80 475 425 355 498 268 311 349 496 205 226 242 472 145 371 196 494 134 350 285 245 163 152 279 153 254 90 2 434 23 410 408 135 147 77 429 298 49 260 58 210 413 154 483 95 19 441 71 14 361 400 249 73 53 266 452 416 74 59 477 423 115 276 182 45 159 64 69 91 493 127 12 487 37 165 191 322 215 356 417 398 338 458 65 117 100 495 421 46 179 131 96 150 500 482 382 280 284 174 252 370 432 344 72 385 489 315 461 424 460 235 88 207 362 31 20 481 21 412 13 85 401 11 290 403 378 278 162 160 342 192 369 288 488 143 146 420 200 323 110 212 16 229 198 60 106 469 490 138 438 128 75 449 332 336 422 392 218 233 353 365 270 319 411 269 415 331 25 476 61 129 273 221 111 308 368 203 499 455 62 89 445 33 437 244 189 228 213 118 271 124 247 329 293 166 234 324 358 341 317 236 123 447 126 359 102 383 459 108 377 241 402 78 414 347 335 30 375 326 104 419 173 307 167 295 443 497 120 343 466 380 204 22 225 318 29 457 386 186 141 277 68 101 310 294 170 339 63 395 256 156 471 222 149 428 40 224 175 168 299 346 79 193 427 309 56 87 202 181 262 92 394 337 272 366 313 133 305 214

fitness functions while designing the evolutionary operators. That is to say, the smaller the fitness value of the optimal individual is, the more the crossover points. With the gradual increase of the fitness value of the optimal individuals in the evolution process, the number of the crossover points is gradually reduced. When the difference between the fitness value and the integer $n(n-1)/2$ is smaller than certain a given positive number, the number of the crossover points is changed to 1. In this way, the evolution speed of the population is accelerated, which gives rise to the improved efficiency of the algorithm. In addition, in order to avoid the local optimum of the programs, the stability of the system requires to be further improved.

References

[1] Dahl, O.J., Dijkstra, E.W. and Hoare, C.A.R. (1972) Structured Programming. Academic Press, London.

[2] Kryszkiewicz, M. (1998) Rough Set Approach to Incomplete Information Systems. *Information Sciences*, **112**, 39-49. http://dx.doi.org/10.1016/S0020-0255(98)10019-1

[3] Kryszkiewicz, M. (1999) Rules in Incomplete Information Systems. *Information Sciences*, **113**, 271-292. http://dx.doi.org/10.1016/S0020-0255(98)10065-8

[4] Stefanowski, J. and Tsoukias, A. (1999) On the Extension of Rough Sets under Incomplete Information. *Proceedings of* 7*th International Workshop on New Directions in Rough Sets*, *Data Mining*, *and Granular Soft Computing*, Physica-Verlag, Yamaguchi, 73-81. http://dx.doi.org/10.1007/978-3-540-48061-7_11

[5] Slowinski, R. and Vanderpooten, D. (2000) A Generalized Definition of Rough Approximation Based on Similarity. *IEEE Transactions on Knowledge and Data Engineering*, **12**, 331-336. http://dx.doi.org/10.1109/69.842271

[6] Holland, J.H. (1992) Adaptation in Natural and Artificial Systems. MIT Press, Cambridge.

[7] Darwin, C. and Burrow, J.W. (2009) The Origin of Species by Means of Natural Selection: The Preservation of Favoured Races in the Struggle for Life. Penguin Classics, USA, 2-3. http://dx.doi.org/10.1017/CBO9780511694295

[8] Darwin, C. (1994) The Descent of Man, and Selection in Relation to Sex. John Murray, London.

[9] Moore, A.J. (1990) The Evolution of Sexual Dimorphism by Sexual Selection: The Separate Effects of Intrasexual Selection and Intersexual Selection. *Evolution*, **34**, 315-331. http://dx.doi.org/10.2307/2409410

[10] Fisher, R.A. (1995) The Evolution of Sexual Preference. *Eugenics Review*, **7**, 184-192.