

应用蚁群算法求解 N 皇后问题

王 宁

(北华大学 吉林 吉林 132011)

摘要: N 皇后问题是一个经典的 NP 搜索问题, 求 N 个皇后的摆放方法, 常见的求解方法为回溯算法和遗传算法。其求解难度随着皇后个数的增长呈指数级增长, 导致求解效率低且系统耗时长。本文通过大量研究, 根据蚁群算法的启发性, 易于结合性和收敛性, 可以快速得到最优解, 将蚁群算法应用到 N 皇后问题中, 提出一种全新的 N 皇后问题求解算法, 大大提高了搜索效率, 减少了最优摆放方法的求解的时间, 节省了系统消耗。

关键词: 蚁群算法; N 皇后问题

中图分类号: TP301.6 **文献标识码:** A **文章编号:** 1009-5349 (2016) 05-0245-02

一、引言

八皇后问题是由著名数学家高斯于 19 世纪提出的, 该问题起源于国际象棋, 指在八乘八的国际象棋棋盘上, 放置八个皇后, 由于国际象棋中, 皇后是可以横向、纵向和斜线移动的, 所有要求八个皇后彼此不能攻击到对方, 即任意两个皇后不能在同一横、纵和斜线上。由于求解八皇后摆放问题需要大量的实验与测试, 所有当时并没有得到所有的解。该问题慢慢扩展演变形成了 N 皇后问题^[1], 即把 N 个皇后摆放在 $N \times N$ 的国际棋盘上, 使任意两个皇后之间不发生冲突。N 皇后问题是一个 NP 难问题, 其求解难度随着 N 的变大, 呈指数增长。^[2]

蚁群算法^[3-6]是由意大利学者 Dorigo 提出, 他根据蚂蚁在觅食时, 通过搜索其他蚂蚁留下的信息, 可以很快找到觅食的最短路径而提出的一种搜索最短路径的算法。蚁群算法具有良好的启发性和易于结合性, 能很好适用于 N 皇后问题求解, 且其收敛性可以得到最优解。本文研究了解决 N 皇后问题的各种算法, 提出一种全新的基于蚁群算法的 N 皇后问题求解算法, 大大提高了路径的搜索效率, 减少了最优摆放方法的求解的时间, 有效提高了问题的解决效率。

二、相关工作

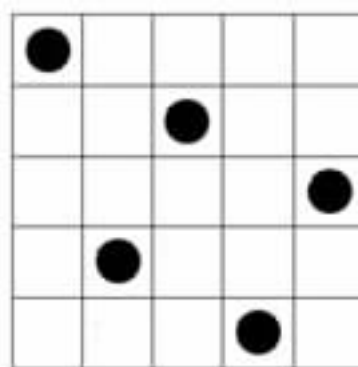
近年来, 许多学者对 N 皇后问题进行了深入研究, 如: 刘寒冰^[7]提出了改进的回溯算法解决 N 皇后问题, 通过改进解空间树和约束条件, 从而减少了求解的计较次数, 提高了求解效率, 得到了可行解。李鸿^[8]经过研究, 提出了一种应用遗传算法解决 N 皇后问题的方法, 通过不断把解进行交叉变异, 逐渐提高解得适应度, 最终得到 N 皇后问题的最优解。上述算法都存在求解速度慢, 且随着 N 的逐渐变大, 求解速度严重降低的情况。针对这一问题, 本文提出一种全新的基于蚁群算法的求解 N 皇后问题的方法, 可以快速有效地得到可行解, 大大提高了求解效率。

三、基于蚁群算法的 N 皇后摆放算法

1. 蚁群建模

设问题中, 皇后的个数为 n , 我们把棋盘的行坐标设为 i , 其中 $1 \leq i \leq n$, 列坐标设为 c_i , 其中 $1 \leq c_i \leq n$ 。根据 N 皇后问题的规则, 每一行只能有一个皇后, C_i 的值代表第 oi 个皇后所摆放的位置, 例如: $C_3 = 5$ 表示将第三个皇后摆放在第三行第五列。因为 N 皇后问题的解不唯一, 设 k 为 N 皇后问题解得个数, $D_k = \{C_1, C_2, \dots, C_n\}$ 为 N 皇后的摆

放方法的集合, 即 N 皇后问题解集合。如当 $n = 5$ 时, 其中为 $D_1 = 13524$ 五皇后问题的其中一种摆法, 如下图所示。



根据 N 皇后问题规则, 任意两皇后不能出现在同一行或同一列, 所以, 当 $i \neq j$ 时, $C_i \neq C_j, i \in [1, n]$ 。设 Jin_e 为蚂蚁 e 的禁行表, 蚂蚁 e 每到一个格子, 将格子信息放入禁行表当中。同时允许表 $Allow_e = \{C_1, C_2, \dots, C_n - Jin_e\}$ 表示蚂蚁 e 下一步允许选择的列数, 蚂蚁 e 每到一个格子, 将 $Allow_e$ 中该蚂蚁所在格子列数删除。应用蚁群算法, 将蚂蚁随机放到第一行的一个位置开始移动, 通过搜索 $Allow_e$ 表的允许移动的格子间的信息素量和格子间的适应度找到最适应的移动路径, 即最优解。如蚂蚁移动路径为 $C_1 - C_2 - C_3$, 则找到的一个解为 $C_1 C_2 C_3$ 。

设 $X(C_i, C_j)$ 为路径 (C_i, C_j) 上的信息素量, 即蚂蚁从 C_i 移动到 C_j 留下的信息素量, 初始时各个路径的信息素量为常数, $X(C_i, C_j) = const$ 。通过蚂蚁的移动, 更新各个格子间的信息素量。为了避免残留信息素过多淹没新产生的信息素, 在所有蚂蚁移动完成后, 对棋盘的信息素进行更新处理, 如下:

$$X(C_i, C_j) = X(C_i, C_j) * v + X'(C_i, C_j) \quad (3.1)$$

式中 v 代表挥发速度, 和现实生活中的气味会随着时间不断扩散挥发一样, 信息素也随着时间逐渐挥发减少。 $X'(C_i, C_j)$ 表示路径 (C_i, C_j) 上新留下的信息素量。

根据皇后的摆放规则可知, 任意两个皇后不能出现在同一斜线上, 即 $|i - j| \neq |C_i - C_j|$, 所以设为 $Z(C_i, C_j)$ 路径 (C_i, C_j) 的适应度, 当满足 $|i - j| \neq |C_i - C_j|$ 且在 Jin_e 中, 所有的 $|i' - j'| \neq |C_i - C_j|, i' \in Jin_e$ 时, $Z(C_i, C_j) = const$, 否则 $Z(C_i, C_j) = const', const > const'$ 。

设 m 为蚂蚁的数量, 蚂蚁 $e = (e = 1, 2, \dots, m)$ 在移动

作者简介: 王宁, 北华大学信息技术与传媒学院助教, 本科, 研究方向: 计算机软件。

过程中, 根据各条路径上的信息素量和适应度决定其移动方向。这里用禁行表 Jin_e 来记录蚂蚁 e 所走过的列数, 即走过的 C_i 的值。蚂蚁在移动的过程中, 蚂蚁 e 从 C_i 移动到 C_j 的概率设为 P_{C_i, C_j}^e , 其表达式如下:

$$P_{C_i, C_j}^e = \begin{cases} \frac{[X(C_i, C_j)]^\alpha \cdot [X(C_i, C_j)]^\beta}{\sum_{s \in Allow_e} [X(C_i, C_s)]^\alpha \cdot [X(C_i, C_s)]^\beta} & \text{若 } j \in Allow_e \\ 0 & \text{否则} \end{cases} \quad (3.2)$$

式中 α 为信息启发式因子^[9], 反映蚂蚁间相互交流的重要程度; β 为期望启发式因子, 反映适应度对蚂蚁选择路径的影响程度。

2. N 皇后摆放算法

Step1 初始化棋盘格子上的信息素量, 初始化禁行表和允许表。

Step2 设置循环次数 $N_0 = 0$, 设置最大循环次数 N_{max} 。

Step3 对蚁群进行初始化, 将 m 只蚂蚁随机放到棋盘的第一行的一个格子上, 更新禁行表和允许表, 初始时 $e = 0$ 。

Step4 根据公式 (3.2) 和禁行表和允许表的信息, 计算蚂蚁 e 移动到下一行每一个允许的列的概率, 选择概率大的列所在的格子移动蚂蚁, 改变蚂蚁所在位置的信息, 更新禁行表和允许表。

Step5 判断蚂蚁 h 是否到达了最后一行, 没有到达跳转至 step4。

Step6 若 $e < m$, $e \leftarrow e + 1$ 并跳转至 step4。

Step7 根据公式 (3.1) 更新棋盘信息素量, 循环次数 $N_0 \leftarrow N_0 + 1$ 。

Step8 若 $N_0 \leq N_{max}$ 则跳转至 step3, 否则皇后摆放结束, 重新初始化棋盘的信息量 $X(C_i, C_j)$ 为常数。

四、实验结果及分析

本文提出了一种应用蚁群算法对 N 皇后问题求解的算法, 在 VC++6.0 环境下用 C++ 语言进行算法编写。这里我们选择了不同数量级的皇后个数做测试用例, 与其他常用算法进行比较。在这里以搜索到的第一个有效解所需要的时间的平均值进行对比, 平均值以十次的运行时间计算, 运行结果如下表所示:

表 1 搜索时间对比 (时间单位: 秒)

皇后数	8	15	30	50	100	500	1000
传统回溯法	0	0.12	481.65	-	-	-	-
文献 ^[8] 遗传算法	0	0	215.23	965.76	-	-	-
本文算法	0	0	0	0.11	0.31	4.32	12.46

对于回溯法求解皇后个数大于 30 和文献^[8]的算法求解皇

后个数大于 50 的问题, 由于所用时间巨大, 本文未能给出求解时间。对于皇后个数小于 15 的情况, 本算法看不出优势, 但当皇后个数大于 30 时, 本算法优势明显, 可以很快求出摆放方法。通过实验结果表明, 将蚁群算法应用到求解 N 皇后问题, 求解速度快, 系统消耗少。

五、结语

N 皇后问题是一个著名的 NP 问题, 通常用遗传算法和回溯算法进行求解。但是回溯算法的时间复杂度为 $O(n!)$, 当 N 变大时, 求解时间呈指数级增长, 遗传算法的求解时间也过长。本文针对传统算法对于求解 N 皇后问题的系统耗时过长的问题, 提出了一种全新的基于蚁群算法的 N 皇后问题求解方法。实验结果表明, 本算法能够快速得到有效解, 提高了求解效率, 减少了系统消耗。当然, 本算法还存在一些不完善的地方, 如不能得到全部有效解, 希望在以后的研究中加以完善。

参考文献:

- [1] 刘娟, 欧阳建权, 陈良军. 用混合遗传算法求解 N 皇后问题 [J]. 湘潭大学自然科学学报, 2007 (06): 37-41.
- [2] 帅训波, 马书南. 应用布尔遗传算子求解 N 皇后问题 [J]. 计算机工程与应用, 2011, 47 (16): 49-68.
- [3] Dorigo M, Maniezzo V, Colomi A. The ant system: Optimization by a colony of cooperating agents [J]. IEEE Transactions on Systems, Man, and Cybernetics - Part B. 1996, 26 (01): 1-13.
- [4] Dorigo M, LM Gambardella. Ant colonies for the traveling salesman problem [J]. Bio systems, 1997, 43 (02): 73-81.
- [5] Dorigo M, Gambardella L M. Ant Algorithms for Discrete Optimization [J]. Artificial Life. 1999, 5 (03): 137-172.
- [6] Dorigo M, Di Caro G. Ant colony optimization: a new meta-heuristic [J]. The Congress on Evolutionary Computation, Washington DC, 1999: 1470-1477.
- [7] 刘寒冰, 李福荣, 叶茂功. N 皇后问题的回溯算法改进 [J]. 软件导刊, 2010 (07): 63-65.
- [8] 陈晓梅. 蚁群算法的原理及其改进 [J]. 广东技术师范学院学报, 2006 (04): 68-70.
- [9] 李鸿. 解决 N 皇后问题的一个遗传算法 [J]. 宿州师专学报, 2000 (04): 85-87.

责任编辑: 杨国栋