DOI: 10.24996/ijs.2021.62.6.33

ISSN: 0067-2904

Meerkat Clan Algorithm for Solving N-Queen Problems

Sura Mazin Ali [1], Noor Thamer Mahmood [2*], Samer Amil Yousif [2]
[1]Political Science College, Mustansiriyah University, Baghdad, Iraq
[2]Computer Science Department, College of Science, Mustansiriyah University, Baghdad, Iraq

Received: 11/9/2020 Accepted: 15/12/2020

Abstract

The swarm intelligence and evolutionary methods are commonly utilized by researchers in solving the difficult combinatorial and Non-Deterministic Polynomial (NP) problems. The N-Queen problem can be defined as a combinatorial problem that became intractable for the large 'n' values and, thereby, it is placed in the NP class of problems. In the present study, a solution is suggested for the N-Queen problem, on the basis of the Meerkat Clan Algorithm (MCA). The problem of n-Queen can be mainly defined as one of the generalized 8-Queen problem forms, for which the aim is placing 8 queens in a way that none of the queens has the ability of killing the others with the use of the standard moves of the chess queen. The Meerkat Clan environment is a directed graph, called the search space, produced for the efficient search of valid n-queens' placement, in a way that they do not cause harm to one another. This paper also presents the development of an intelligent heuristic function which is helpful to find the solution with high speed and effectiveness. This study includes a detailed discussion of the problem background, problem complexity, Meerkat Clan Algorithm, and comparisons of the problem solution with the Practical Swarm Optimization (PSO) and Genetic Algorithm (GA. It is an entirely review-based work which implemented the suggested designs and architectures of the methods and a fair amount of experimental results.

Keywords: Swarm Intelligence, N-Queen problem, MCA, PSO, GA, NP

خوارزمية زمرة القط الافريقي لحل مشكلة N-Queen

سرى مازن علي [1], نور ثامر محمود [2]*, سمر اميل يوسف[2]
[1] كلية العلوم السياسية, الجامعة المستنصرية, بغداد, العراق
[2] قسم علوم الحاسوب, كلية العلوم, الجامعة المستنصرية, بغداد, العراق

الخلاصة

يشيع استخدام ذكاء السرب والأساليب التطورية من قبل الباحثين في حل القضايا الصعبة التوافقية من نوع NP (غير الحتمية متعددة الحدود). يمكن تعريف مشكلة N-Queen على أنها مشكلة اندماجية أصبحت مستعصية على الحل بالنسبة لقيم n الكبيرة، وبالتالي، يتم وضعها في مشاكل فئة NP. في هذا البحث تم اقتراح حل لمشكلة N-Queen باستخدام خوارزمية زمرة القط الافريقي (Meerkat Clan Algorithm). يمكن تعريف مشكلة n-Queen بشكل أساسي على أنها واحدة من أشكال مشكلة 8-Queen العمومية، والتي يكون الهدف فيها وضع 8 ملكات، بطريقة لا تمتلك أي من الملكات القدرة على قتل أخرى باستخدام

*Email: noorthamer@uomustansiriyah.edu.iq

معيار تحركات ملكة الشطرنج. بيئة زمرة القط الافريقي عبارة عن رسم بياني موجه يسمى مساحة البحث يتم إنتاجها من أجل البحث الفعال عن مواضع n–queens الصالحة بطريقة لا تسبب ضرراً لبعضها البعض. يقدم هذا البحث أيضاً تطوير وظيفة إرشادية ذكية تساعد في إيجاد الحل بسرعة وفعالية عالية. تتضمن هذه الدراسة مناقشة تفصيلية لخلفية المشكلة، وتعقيد المشكلة، وخوارزمية زمرة القط الافريقي) أي ذكاء الاسراب ومقارنة حل المشكلة مع خوارزمية التحسين العملي للسرب(PSO) والخوارزمية الجينية، وتجري مقارنة بين هذه الخوارزميات الثلاثة. هذا العمل قائم على المراجعة بالكامل ونفذ التصميم والبنية المقترحة للطريقة وكمية لا بأس بها من النتائج التجريبية.

Introduction

A swarm can be defined as a large amount of simple and homogenous factors that locally interact amongst one another, and with their environment, without any central regulation for allowing the emergence of a general interesting behavior. The swarm-based methods have lately emerged as a set of the population-based, nature-inspired methods which can produce fast, low cost, and robust solutions to a number of complicated tasks [1-3].

The MCA is efficiently dependent upon the behavior and parameters of Meerkats, which are social mammals, living in groups of 5 to 30 individuals. Due to the fact that they are amiable animals, they share latrine as well as the parental care duties. Each crowd has one commanding alpha male and an overwhelming alpha female, as well as a specified domain where they change in the case where the nourishment is rare or they are constrained out by a more grounded crowd. This algorithm begins by fixed steps, which are utilized to initialize the parameter values, and flows with the number of the iterated steps for discovering the optimal solution to the problem [4].

The N-Queen problem is represented by placing ‘n’ queens on a chessboard in a way that none of the queens has the ability of killing the rest with the use of the standard moves of the chess queen. There can be named 3 such cases where the queen has the ability of killing another one [5];

a. in the case where 2 queens are in one row,

b. in the case where 2 queens are in one column,

c. the case where one of the queens is in the diagonal of another one.

N-Queen Problem

The 8 queens problem is the famous NP-complete problem which was firstly introduced in 1850 by C. F. Gaus [6]. This Problem has been researched by a number of mathematicians in the 19th century. The main challenge of the problem is the fact that it needs numerous calculations. The general problem of the N-Queen was discussed by Yaglom and Yaglom in the 1950s [6]. The general problem of the N-Queen has been characterized by the following constraints into an NxN grid with following conditions [6]:

1.Only 1 queen may be put in row.

2.Only 1 queen may be put in column.

3.Only 1 queen may be put on diagonal.

4.Precisely N queens have to be put on that grid.

There has been a number of methods that has been introduced to address this problem, such as the program development, algorithmic design, AI, and parallel and distributed computing. Such wide-spread interest in the N-Queen problem is partly a result of the characteristics which identifies complex problems, viz. which satisfies a group of global constraints. In the section below, the proposal of the molecular algorithm to solve that problem will be introduced [6].

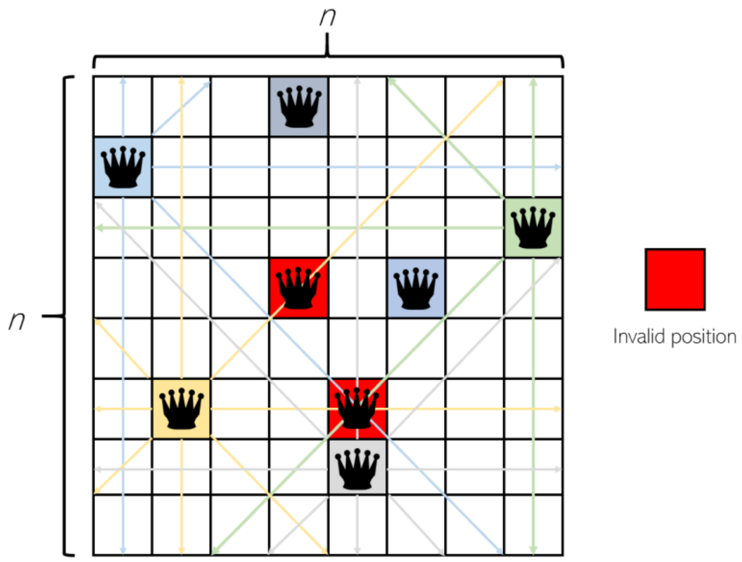

Figure 1-Almost a solution of the 8-queens' problem [7].

Also, the N-Queens is considered as one of the NP-complete problems. There are several metaheuristic (swarm intelligence and nature-inspired) algorithms that have been used to solve the NP-complete problems, such as the bees algorithm, camel herd algorithm, ant colony algorithm, harmony search, scatter search, and so on [8-11].

Related Work

As of late, the region of Evolutionary Computation has made its mark. Two of the common advanced approaches are the GAs and PSO, both of which being utilized in optimal problems. While the two methodologies are upheld to discover an answer for a given target work, they utilize various systems and computational endeavours. Consequently, it is worthy to evaluate their usage.

Martinjak *et al*. [2007] showed that the mathematical problem of the n-queens may be successfully solved with the use of heuristic methods, even in the cases of very large problem dimensions. The heuristic algorithm was carried out for the simulation of the Tabu search, simulated annealing, and GA algorithms by comparing their levels of efficiency. The results showed that a conceptually quite simple heuristic function (like a case in which the neighborhood includes n-tuples differed from current solution in the positions of the two queens) might address this difficult non-deterministic polynomial problem [12].

Khan *et al*. [2009] suggested an effective solution by operating an 8-queen problem that can easy extends to large 'n' values due to the simplistic model of search space. They found that their solution can be effective for the 8-queen in the case where the parameters have been set as swarm size = 15, alpha = 1.50, and beta = 1, or alpha = 1, beta = 1.50. It was concluded that the ACO has the ability of providing a sufficient solution in a reasonable duration for the problems of combining optimizations. As a future work, they proposed exploring its applicability to a number of other equivalent problems [5].

Sadiq *et al*. [2010], considered the N-Queens problem as a very difficult one, which several researchers were interested in solving with a variety of intelligent approaches. Their study presented a new hybrid technique to solve the N-Queen problem. The proposed method depended on DNA-computing and Tabu search algorithm. Tabu search is used as a tool for increasing the efficiency of the system by reducing the run time and memory capacity as well as the number of the random generated states. The experimental results demonstrated that the use of the hybrid approach provided the best results when compared with the classical DNA-computing approach. The proposed method produced results when

all solutions with size 11-by-11, and multiple solutions with size 20-by-20 as maximum. The run time of the proposed system was lower than that of the classical DNA-computing by 50% and the reduction in memory capacity was 75% [13].

Sadiq *et al*. [2013] solved the N-queen's problem using three swarm intelligence algorithms, namely the Bees algorithm, particle swarm optimization (POS), and Cuckoo search. The study showed that these algorithms provided relatively better results. By comparison, the Bees algorithm was more efficient than the PSO, which performed better than the Cuckoo search to handle n-queens problem [14].

Al-Obaidi *et al*. [2018] presented the Meerkat Clan Algorithm and displayed its ability for solving the problem of the Traveling Salesman. The MCA results were obtained through the division of the set of the solutions into 2 groups (care and foraging). The majority of the operation was carried out on the foraging group and the poorest solutions were substituted by the optimal ones in the care set. The poorest solution in the care set was dropped and another randomly created solution was added. Their results demonstrated the exceptional performance of the capacity of the algorithms in obtaining the optimum or near-optimum solutions at quite fast rates [4].

Nowadays, the computer science community is aware of the importance of the development of solutions for the complicated problems. As discussed above, learning about the collective behaviors of the living creatures can be considered as a motivation to study and provide a valuable swarm-based meta-heuristics analysis tools. The efforts until now have shown the possibility of those new approaches to find successful solutions to a wide range of applied optimization problem types. Actually, there is no single 'best' approach. A variety of applications have been more satisfactory for solving various problems, which either results in more adequate solutions, or improves the speed. In addition to that, the specific approach's suitability does not only rely on the issue; various processes are more appropriate for various individuals, according to their capabilities and knowledge [4].

MCA to Solve N-Queen Problem

Meerkats Metaheuristic

Meerkats can be described as dwelling mammals which live in massive open networks with a variety of entrances, where they are only capable of leaving during the day [4]. Meerkats are very friendly animals which occupy abandoned domains in the form of clusters, usually of about 10 to 30 individuals, even though considerably larger groups are normally seen in zones of a sufficient supply of food. The group usually includes 3-4 female and male match families, as well as their offspring. After being a wake early in the morning, they rise out of the tunnels for sunbathing. The majority of the group will head out for rummaging for sustenance whereas the rest will either play the roles of gate-keepers or keep an eye on the young. By standing straight on the tails and the rear legs on the highest hill point and in the shrubs, the monitor Meerkats may be having a good advantage to call attention for keeping an eye out for the moving towards the predators, particularly from the sky. A progression of a variety of the caution calls are going to be sounded afterwards for alarming the rest of the gathering about the approaching risk, often making the whole group plunge in the underground tunnel for stowing away [4].

The Artificial Model of the Meerkat Clan

This section will present the Meerkat Clan Algorithm which is dependent upon the behaviour and the parameters of the Meerkats, along with the impacts of those parameters on the efficiency of the algorithm. The significant parameters were obtained from the behaviours of the Meerkats and employed in the MCA, which include the size of the clan n, care group size c, foraging group size m, number of neighbours K, and worst foraging and care ratio, Fr and Cr, respectively. This algorithm begins by specific steps which are utilized to initialize the parameter values and flow with the amount of the iterated steps for the discovery of the optimum solution to the problem. The specified steps which have been utilized in this algorithm for any problem are:

1. Initialization of the values of the parameters.
2. Generation of the random clan.
3. Computation of the clan fitness.
4. Selection of the optimal solution as the Sentry.
5. Dividing the clan to 2 groups, i.e. care and foraging (c and m, respectively).

The iterated processes which can differ based on the type of the problem are:

1. Generation of the random neighbours for every one of the foraging solutions.
2. Swapping the optimal neighbour with the foraging solution in the case where it is better.
3. The poorest solutions for the foraging group will be swapped with the optimal solutions in the care group.

4. Dropping the poorest care solutions and generating new ones in a random manner.

```
Parameter
        n       clan size
        m       foraging size     where m < n
        c       care size         n-m-1
        Fr      worst foraging rate
        Cr      worst care rate
        k       neighbor solution

Begin
        Generate random clan of solutions clan(n)
        Compute fitness for clan solutions
        Sentry= best solution of clan
        Divide the clan into two groups (foraging & care)
        While not termination condition Do
                For i=1 to m
                        Generate k neighbors from foraging set
                        foraging(i)= best one from k neighbor
                end for
                Swap the worst for Fr solution in foraging group with best ones solution in care
                group;
                Drop the worst Cr solution from care group and generate ones solution randomly;
                Select the best one of foraging call it best_forg
                If best_forg <= Sentry then
                        Sentry ←→ best_forg
                end if
        end while
End
```

Figure 2-Meerkat Clan Algorithm Pseudocode [4].

From the earlier explanations about the Meerkat animal-inspired MCA, the followings are the steps for the MCA to solve the N-Queen problem:

Initialization the parameters;
Creating clan of boards (clan) randomly;
Calculating the fitness for all boards in clan;
Choosing the optimal board as 'sentry';
Dividing the boards (clan) to 2 groups (care and foraging);
Repeat

- Generating the neighbors for the foraging group using neighbor generation strategy;
- Updating the foraging group;
- If there is a board that is better than the sentry, then Sentry = the best board;
- Swapping the worst boards in the foraging group with optimal boards in the care group;
- Dropping the worst boards in the care group and generating other boards in a random manner;

Until (Goal is found or Max. Gen.);
Solution = Sentry;

Experimental Results

We focused on evolutionary and swarm intelligent algorithms. In addition to our proposed algorithm for solving N-Queen problem by Meerkat algorithm, some new research has introduced two important algorithms, namely the GA [15] and the PSO [15]), which were utilized for solving the N-Queens problem. in the results of the tested parameters for the MCA, GA, and PSO algorithms are listed in Tables 1, 2 and 3, respectively.

Table 1- MAC parameters for solving the N-Queens Problem

Initializations	**Random**
Representation	Integer String which is n long
Clan size	**100**
Forging sixe	**75% of clan**
Care size	**24% of clan**
Worst foraging rate	**1% of foraging**
Worst care rate	**1% of care**

Table 2- GA's operators and parameters to solve the N-Queens Problem

Initializations	**Random**
Representations	Integer String which is n long
Selection	Selection of the Tournament
Recombination	Partial Matched Crossover (PMX)
Mutation	Swap
Likelihood of the Mutation	0.50
Size of the Population	10, 20, 30,…, 100
Maximal Number of the generations (NoG)	100
Stop Condition	Solution or Number of the Generations= NoG

Table 3- Description of the PSO parameters to solve the N-Queens Problem

Initializations	**Random**
Representation	Integer String of Length= n
w	w= ((Tmax - G) * (00.90 - 00.40) / Tmax) + 00.40
c1&c2	2.0
r1&r2	Random [0..1]
Swarm Size or Number of Particles	10, 20, 30,…, 100
Maximum Number of Iterations(Tmax)	100, 30000
Stopping Condition	Solution or Number of iterations= Tmax

For 10 times running and for 10-100 as a population size, the mean value of the optimal results for 8-queens have been presented. Table (4) shows that the MCA provides better solutions for solving the 8-Queens problem as compared to the GA and the PSO, in terms of time and number of generations.

Table 4- Best generation and consumed time to find 8-queens problem solution in MCA, GA, and PSO.

Pop. Size	**MCA**		**GA [16]**		**PSO [16]**	
	Best Gen.	**Time (Sec.)**	**Best Gen.**	**Time (Sec.)**	**Best Gen.**	**Time (Sec.)**
10	5	0.028	12	1.8485	13	1.1977
20	2	0.013	7	1.7053	4	1.3741
30	1	0.017	3	1.5430	2	1.5269
40	2	0.027	3	1.3966	4	1.4405
50	2	0.014	5	1.4659	3	1.3814
60	1	0.032	2	1.4373	3	1.4725
70	1	0.022	2	1.9066	4	1.6210
80	2	0.020	3	1.9255	3	1.8541
90	1	0.017	3	2.3777	4	2.2561
100	1	0.019	3	2.4925	3	2.1666

For 16-queens, Table (5) illustrates the results of best generations and consumed time for MCA, GA, and PSO. The best algorithm is the MCA, which was superior to the other ones by a large margin.

Table 5- Best iteration and consumed time to find 16-Queens Problem Solution in MCA, GA and PSO,

Pop. Size	MCA		GA [16]		PSO [16]	
	Best Iter.	Time (Sec.)	Best Iter.	Time (Sec.)	Best Iter.	Time (Sec.)
10	20	0.029	60	1.9999	23547	61.9999
20	24	0.049	70	2.9999	18444	95.9999
30	17	0.053	39	2.3333	10003	76.9999
40	14	0.061	31	3	143366	147.6666
50	14	0.069	25	2.3333	15150	194.6665
60	10	0.105	24	2.6666	13247	203.9999
70	8	0.061	35	4.6666	3853	69.6666
80	8	0.067	36	4.9999	4443	91.3333
90	8	0.096	30	4.6666	7160	167.3333
100	6	0.100	39	6.6666	4548	117.6666

Tables 4 and 5 show that the MCA obtained the best results in a very good time and best iteration numbers. Also, GA has a good result, whereas the worst algorithm is PSO in terms of both consumed time and iteration numbers.

MCA required less iteration, as shown in Table 5. Also, the frequency was better than that obtained by GA and PSO, because of the higher divergence of Meerkat. The strategic neighborhood adopted by the Meerkat is one of the reasons for this diversification.

The average number of iterations required by the MCA to find the solution is less those required by GA (about 66%) and PSO (about 92%).

Conclusions

N-Queens problem was selected for comparing the performance of MCA against both GA and PSO. MCA showed simple behavior and operators, its performance under different cases is well in small and large search space size. The experimental results showed that MCA was clearly superior compared with GA and PSO. This result was achieved because of the good diversity and exploration features of this algorithm which depend on the neighbours-generation strategy. As a future work, the order crossover operation can be used as a type of neighbourhood generation. Specifically, this can be performed for those operators with several crossover methods; for example, the Very Greedy Crossover, Cycle Crossover, and Crowding Crossover.

Acknowledgments

The authors would like to show gratitude to the Mustansiriyah University (www. uomustansiriyah.edu.iq), Baghdad-Iraq, for its appreciated support to the present work.

References

1. Panigrahi B. K., Shi Y., Lim M.-H. (eds.). **2011**. *Handbook of Swarm Intelligence. Series: Adaptation, Learning, and Optimization*, Vol 7, Springer-Verlag Berlin Heidelberg, ISBN 978-3-642-17389-9.
2. Blum C., Merkle D. (eds.). **2008**. *Swarm Intelligence – Introduction and Applications. Natural Computing* , Springer, Berlin.

3. Hussein S. A. and Sadiq A. T. **2019**. An Improved Crow Search Algorithm for Solving the Traveling alesman Problem, *Journal of Computational and Theoretical Nanoscience*, 16(3): 978-981.
4. Sadiq A. T., Abdullah H. S., Ahmed Z. O. **2018**. *Meerkat Clan Algorithm: A New Swarm Intelligence Algorithm*, *Indonesian Journal of Electrical Engineering and Computer Science*.
5. Khan S., Bilal M., Sharif M., Sajid M., Baig R. **2009**. Solution of n-Queen Problem Using ACO, *IEEE*, 978-1-4244-4873-9/09.
6. Ahrabian H., Merzaei A., Nowzari-Dalini A. **2008**. A DNA Sticker Algorithm for Solving N-Queen Problem, *International Journal for Computer Science and Applications*, **5**(2): 12-22.
7. Russell S., Norvig P. **2003**. *Artificial Intelligence: A Modern Approach*, chapter 3. Prentice Hall, 2nd edition.
8. Mohammed R. A., Duaimi M. G., Sadiq A. T. **2017**. Modified Bees Swarm Optimization Algorithm for Association Rules Mining, *Iraqi Journal of Science*, **58**(1B): 364-376.
9. Sadiq A. T. **2013**, Harmony-Scatter Search to Solve Travelling Salesman Problem, *Iraqi Journal of Science*, **54**(4): 965-974.
10. Sadiq A. T., Abdullah H. S., Ahmed Z. O. **2017**. Camel Herds Algorithm: a New Swarm Intelligent Algorithm to Solve Optimization Problems, *International Journal on Perceptive and Cognitive Computing*, **3**(1).
11. Yousif R. *and* Sadiq A. T. **2013**. "An Improved Subset AntNet Algorithm for Numerical Optimization of the Shortest Path Problem", *International Journal of Scientific and Engineering Research* , **4**(7).
12. Martinjak I., Golub M.. **2007**. Comparison of Heuristic Algorithms for the N-Queen Problem, 29th International Conference on Information Technology Interfaces, IEEE, Cavtat, Croatia, ISSN**:** 1330-1012.
13. Sadiq A.T., Abdullah H. S., Fadhel M.N. **2010**. Proposal of Tabu DNA Computing Algorithm to Solve N-Queens Problem, 2010 Conference first conference in computer science department in university of technology of Baghdad.
14. Sadiq A. T. and Younis M. T. **2013**. Using Swarm Intelligence Algorithms to Solve n-Queens Problem, *Al- Mustansiriyah J. Sci*. **24**(5).
15. Mouser C.R., Dunn S.A. **2005**. Comparing Genetic Algorithms and Particle Swarm Optimization for Inverse Problem Exercise, *Austral. Mathematical Soc*.
16. Hammad A. W., Thannoon B. N .**2007**. Genetic Algorithm Versus Particle Swarm Optimization in N-Queen Problem, *Journal of Al-Nahrain University*, **10**(2):172-177.