MINAR International Journal of Applied Sciences and Technology

Article type : Research Article

Date Received : 01/10/2020
Date Accepted : 22 /10/2020
Date published : 01/12/2020

: www.minarjournal.com

http://dx.doi.org/10.47832/2717-8234.4-2.9

SOLVING N-QUEEN PROBLEM USING PROBABILITY COLLECTIVE

Lutfia Khalifa Haj MOHAMED[1]

Abstract

Many types of research solve N-Queen Problem by using various techniques as Genetic algorithm (GA), particle swarm optimisation (PSO), and simulating annealing (SA). This paper motivates and describes the use of probability collectives (PC) with coordination multi-agent system to solve the N-Queen Problem. The main challenge is to make the agents work in a coordinate a way, optimising the local utilities and contributing the maximum towards optimisation of the global objective.
Keywords: Probability Collectives, Collective Intelligence, Multiagent systems, N-Queen Problem..

[1] Elmergib University, Libya, lutfiahaj@gmail.com, https://orcid.org/0000-0003-0793-9793

Literature Review

In a few past years, there are challenges to solve complex problems by using a set of distributed intelligence agents. These agents act in some specific direction to find local payoffs or the best solution. Hence, Probability collectives (PC) in the framework of CION are a new distributed optimisation algorithm that was first introduced by Dr David Worlpert in 1999 in a Technical Report suggested to NASA [6] [9]. In 2004, many modifications and applications have already been evolved, where Lee and Worlpert change the word utility with a private utility to reduce the sample size utilised in the PC algorithm with no prejudice and low contrast. The sample size has effectively been reduced using the data aging technique (Beneniawski et al 2005). Kulkarni et al (2008) shorted the sample region around the present optimal points by modifying the sample principle on original MontE-Carlo. In 2011, it has been updated some of the strategies like Steepest Decent, Nearest Newton, and Brower Fixed Point method by Worlpert et al [10]. Moreover, Worlpert et al used the significance sampling and parametric Machine Learning technique to classify the PC algorithm as 'Delayed Sampling' and place forward another 'Immediate Sampling'. Kulkarni et al. (2011) have been used the Broyden Fletcher Goldfarb Shanno method (BFGS) to optimise the model [10][9].

Some application is implemented using the approach of the PC to solve different areas. The benchmark problems demonstrate that the PC algorithm outperforms on GA in the rate of decent, trapping in fake minima and long term optimisation (Hung et at 2005). PC is also used to evaluate the queens, bar, and bin packing problems by (Worlpert et al). It has been solved as multi-depot multiple travelling salesmen problems (Kulkarni et al 2010; Kulkarni et al 2010b), the feel assignment problem (Antoine et al 2004), university course timetabling problems (Autory and Brian 2008), and vehicle routing problems (Kulkarni et al 2010) [10].

1. Introduction to probability collective

PC algorithm is a modern method to solve distributed optimisation problems and the approach of PC which is in the framework of CION with its linkages to Game Theory, Statistical Physics, and Optimisation [11][1]. In PC, the variables are denoted as individual agents/players and the distributed optimisation problem is considered as a game played by these agents [2] [4]. Theory of PC allocates probability distributed values to select the agents' moves and allowing for each agent to autonomously update its own probability distribution at each iteration. These agents select a particular action based on the highest probability to optimise their own utility function. Thus, The algorithm continues to find the best solution until the convergence reaches to the global solution or one of stopping criteria such as $T = 0$ [5].

In some applications, the agents require knowing the inter-agent-relationship. It is one of the strategies set which each agent assumed to realise. This permits all the agents to define the correct access to the strategy collections. These decisions are picked independently by every agent finding the available information due to optimise the local utility and realise the global utility [5].

1.1 Advantages of Probability Collectives

Probability collectives have many benefits over the other techniques that can use optimisation tasks:

In PC, every agent autonomously updates its own probability values at any time, and it can be used on continuous, discrete or mixed variables [5] [3].

- A set of probability strategy has always been a vector of real numbers with this way it permits the technique of simple optimisation for Euclidean vectors as the gradient descent to be usable [3].
- The cost function of the PC can be irregular or noisy because it is a robust algorithm [3].
- A variable with a peaky distribution plays a more significant role in the optimisation task than a variable with a broad distribution since PC provides the sensitivity information about the problem [4].
- Each agent (variable) can find the minimum value of the global objective by using the Homotopy function [6].

1.2 N-Queens Problem

The N-queens problem has been proposed by Max Bezzel in 1848 for normal 8×8 chessboard, which belongs to the class of constraint satisfaction problems. The objective of the problem is to put N queens on a $N \times N$ chessboard where there are no conflicts among any of queens such as no shared rows, columns, diagonals [8]. This problem is formalized as follows.

- Let $V = \{v_1, \dots \dots . . v_n\}$ are a group of variables and each of them is identical to a row in the chessboard [8].
- N are a number of queens.
- Every variable v_i takes a value from the domain where $D_i = \{1,2,3 \dots \dots \dots \dots n\}$, where every v_i corresponds to a column of the chessboard which can place a queen [7].
- The constraints are $v_i \neq v_j, v_i - v_j \neq i - j, v_i - v_j \neq j - i$ [7].
- The objective function is non-attacking queens on the $N \times N$ chessboard by considering the chess rules.

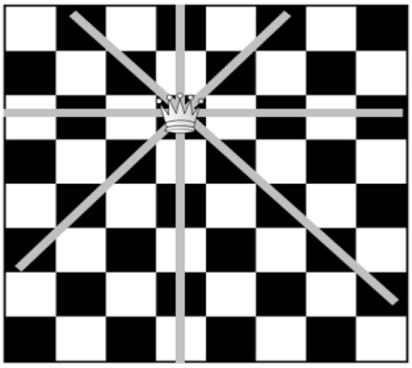


Figure 1: Example of Queen's Move

2. probability Collectives Algorithm

Probability collectives can formalise as a group of N agents, each agent x_i can take on a finite number of values from interval $\Omega \in [x_i^L, x_i^H]$ and builds a set of solution through a strategy set x represented as [5][9]:

$$X_i = \{X_i^{[1]}, X_i^{[2]}, \dots\dots\dots, X_i^{[m_i]}\}, \quad i \in \{1,2,\dots\dots.N\} \qquad (1)$$

Where m_i is the number of strategies and N is the number of variables, In PC, each agent combines the strategy Y_i^r which choses randomly by other agents as:

$$Y_i^{[1]} = \{X_1^{[?]}, X_2^{[?]}, \dots\dots, X_i^{[1]}, \dots\dots\dots, X_{N-1}^{[?]}, X_N^{[?]}\} \qquad (2)$$

The superscript [?] denotes to random selection and each agent has formed one strategy set for every of the residual strategies. Accordingly, the set of solutions build by agent i as shown below.

$$\begin{aligned} Y_i^{[2]} &= \{X_1^{[?]}, X_2^{[?]}, \dots\dots, X_i^{[2]}, \dots\dots\dots, X_{N-1}^{[?]}, X_N^{[?]}\} \\ Y_i^{[3]} &= \{X_1^{[?]}, X_2^{[?]}, \dots\dots, X_i^{[3]}, \dots\dots\dots, X_{N-1}^{[?]}, X_N^{[?]}\} \\ Y_i^{[r]} &= \{X_1^{[?]}, X_2^{[?]}, \dots\dots, X_i^{[r]}, \dots\dots\dots, X_{N-1}^{[?]}, X_N^{[?]}\} \\ Y_i^{[m_i]} &= \{X_1^{[?]}, X_2^{[?]}, \dots\dots, X_i^{[m_i]} \dots\dots, X_{N-1}^{[?]}, X_N^{[?]}\} \end{aligned} \qquad (3)$$

As the same, all the residual agents form their collective strategy sets as shown in equation (3), then every agent i evaluates the objective function for each their combined strategy set $Y_i^{[m_i]}$ as:

$$\left[G\left(Y_i^{[1]}\right), G\left(Y_i^{[2]}\right), \dots\dots, G\left(Y_i^{[r]}\right), \dots\dots\dots, G\left(Y_i^{[m_i]}\right)\right] \qquad (4)$$

Each agent finds the sum of the objective function for its combined strategy set to be minimised as follows [9]:

$$\begin{array}{lll} Y_1^{[1]} = \{X_1^{[1]}, X_2^{[?]}, \dots\dots, X_i^{[?]}, \dots\dots\dots, X_{N-1}^{[?]}, X_N^{[?]}\} & G\left(Y_1^{[1]}\right) & \\ Y_1^{[2]} = \{X_1^{[2]}, X_2^{[?]}, \dots\dots, X_i^{[1]}, \dots\dots\dots, X_{N-1}^{[?]}, X_N^{[?]}\} & G\left(Y_1^{[2]}\right) & \sum_{r=1}^{m_i} G(Y_1^{[r]}) \\ Y_1^{[m_i]} = \{X_1^{[m_i]}, X_2^{[?]}, \dots, X_i^{[?]}, \dots\dots\dots, X_{N-1}^{[?]}, X_N^{[?]}\} & G\left(Y_1^{[m_i]}\right) & \\ Y_i^{[1]} = \{X_1^{[?]}, X_2^{[?]}, \dots\dots, X_i^{[1]}, \dots\dots\dots, X_{N-1}^{[?]}, X_N^{[?]}\} & G\left(Y_i^{[1]}\right) & \\ Y_i^{[2]} = \{X_1^{[?]}, X_2^{[?]}, \dots\dots, X_i^{[2]}, \dots\dots\dots, X_{N-1}^{[?]}, X_N^{[?]}\} & G\left(Y_i^{[2]}\right) & \sum_{r=1}^{m_i} G(Y_i^{[r]}) \\ Y_i^{[m_i]} = \{X_1^{[?]}, X_2^{[?]}, \dots, X_i^{[m_i]}, \dots\dots\dots, X_{N-1}^{[?]}, X_N^{[?]}\} & G\left(Y_i^{[m_i]}\right) & \\ Y_N^{[1]} = \{X_1^{[?]}, X_2^{[?]}, \dots\dots, X_i^{[?]}, \dots\dots\dots, X_{N-1}^{[?]}, X_N^{[1]}\} & G\left(Y_N^{[1]}\right) & \\ Y_N^{[2]} = \{X_1^{[?]}, X_2^{[?]}, \dots\dots, X_i^{[?]}, \dots\dots\dots, X_{N-1}^{[?]}, X_N^{[2]}\} & G\left(Y_N^{[2]}\right) & \sum_{r=1}^{m_i} G(Y_N^{[r]}) \quad (5) \\ Y_N^{[m_i]} = \{X_1^{[?]}, X_2^{[?]}, \dots, X_i^{[?]}, \dots\dots\dots, X_{N-1}^{[?]}, X_N^{[m_i]}\} & G\left(Y_N^{[m_i]}\right) & \end{array}$$

It is a very hard to find the minimum of function $\sum_{r=1}^{m_i} G(Y_i^{[r]})^n$, because there are several possible local minima. For this reason, the objective function $\sum_{r=1}^{m_i} G(Y_i^{[r]})^n$ are converted into another to topological space by building an easier function E and placing it in a new form known as a Homotopy Function.

$$J_i(q(x_i), T) = \sum_{r=1}^{m_i} G\left(Y_i^{[r]}\right) - T * E \quad , \qquad T \in [0, \infty) \qquad (6)$$

For all agents are assigned uniform probability distribution $q(X_i^{[r]})$ such as if the number of strategies $m_i = 5$ then each agent i will take uniform probability distribution $q\left(X_i^{[r]}\right) = \frac{1}{5}$ that is shown in figure 2.

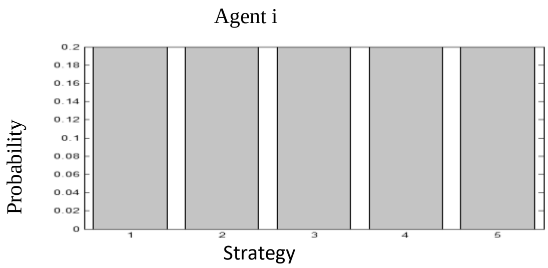


Figure 2: Uniform Probability Distribution $m_i = 5$ of Agent I

Hence, $q\left(X_i^{[r]}\right)$ is calculated as:

$$q\left(X_i^{[r]}\right) = \frac{1}{m_i} \quad , k = 1,2,3 \ldots\ldots\ldots, m_i \qquad (7)$$

Every agent calculates the expected utility function $\sum_{r=1}^{m_i} E\left(G(Y_i^{[r]})\right)$ through using a joint product probability which is formed from the probability distribution of other agents randomly, The expected objective function for N agents are as follows [2][9]:

$$\left.\begin{aligned} &G\left(Y_1^{[1]}\right) q\left(X_1^{[1]}\right) \prod_{(1)} q\left(X_{(1)}^{[?]}\right) = E\left(G\left(Y_1^{[1]}\right)\right) \\ &G\left(Y_1^{[r]}\right) q\left(X_1^{[r]}\right) \prod_{(1)} q\left(X_{(1)}^{[?]}\right) = E\left(G\left(Y_1^{[r]}\right)\right) \\ &G\left(Y_1^{[m_1]}\right) q\left(X_1^{[m_1]}\right) \prod_{(1)} q\left(X_{(1)}^{[?]}\right) = E\left(G\left(Y_1^{[m_1]}\right)\right) \end{aligned}\right\} \sum_{r=1}^{m_1} E\left(G\left(Y_1^{[m_1]}\right)\right)$$

$$\left.\begin{aligned} &G\left(Y_N^{[1]}\right) q\left(X_N^{[1]}\right) \prod_{(N)} q\left(X_{(N)}^{[?]}\right) = E\left(G\left(Y_N^{[1]}\right)\right) \\ &G\left(Y_N^{[r]}\right) q\left(X_N^{[r]}\right) \prod_{(N)} q\left(X_{(N)}^{[?]}\right) = E\left(G\left(Y_N^{[r]}\right)\right) \\ &G\left(Y_N^{[m_N]}\right) q\left(X_N^{[m_N]}\right) \prod_{(N)} q\left(X_{(N)}^{[?]}\right) = E\left(G\left(Y_N^{[m_N]}\right)\right) \end{aligned}\right\} \sum_{r=N}^{m_N} E\left(G\left(Y_N^{[r]}\right)\right) \qquad (8)$$

Now, we need to replace E used in the Homotopy function and put its place a Convex function such as the Entropy Function [6][5].

$$S_i = -\sum_{r=1}^{m_i} \left[q\left(X_i^{[r]}\right) \log_2 q\left(X_i^{[r]}\right)\right] \qquad (9)$$

Hence, each agent i minimized its Homotopy function as:

$$J_i\left(q\left(X_i^{[r]}\right), T\right) = \sum_{r=1}^{m_i} E\left(G(Y_i^{[r]})\right) - T * S_i$$

$$= \sum_{r=1}^{m_i} E\left(G(Y_i^{[r]})\right) - T * \left(-\sum_{r=1}^{m_i} \left[q\left(X_i^{[r]}\right) \log_2 q\left(X_i^{[r]}\right)\right]\right) \qquad (10)$$

Where $T \in [0, \infty)$

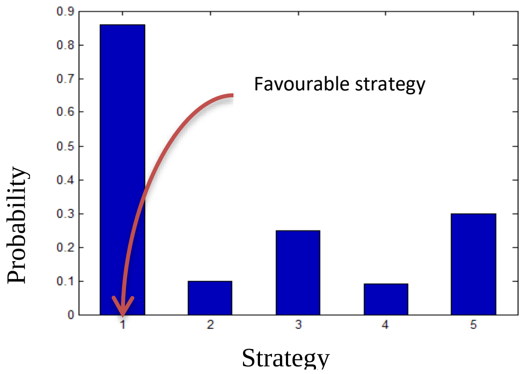
Some of a suitable optimisation technique is used to find the minimization of Homotopy function such as Nearest Newton Descent Scheme (NNDS), Broden-Flectcher-Goldfarb-Shanno (BFGS) and Deterministic Annealing (DA) (Kulkarni et al 2015)[6][9]. Thus, the NNDS will be used in this thesis to update the probability of all the strategies of each agent i as follows:

$$q\left(X_i^{[r]}\right) \leftarrow q\left(X_i^{[r]}\right) - \propto_{step} * q\left(X_i^{[r]}\right) * K_{r\ update} \qquad (11)$$

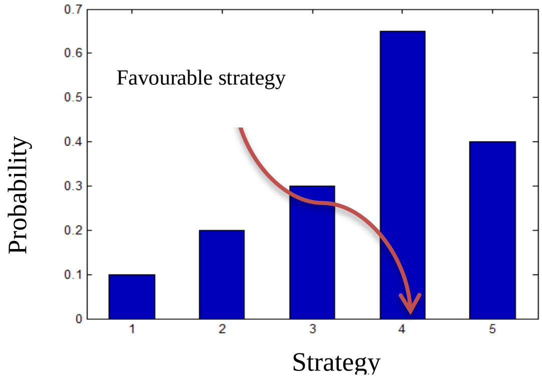
Where $K_{r\ update} = \frac{\text{Contribution of agent } i^r}{T} + s_i(q) + \ln\left(q\left(X_i^{[r]}\right)\right) \qquad (12)$

And $\text{Contribution of agent } i^r = E\left(G\left(Y_i^{[r]}\right)\right)_n - \left(\sum_{r=1}^{m_i} E\left(G\left(Y_i^{[r]}\right)\right)\right)_n \qquad (13)$

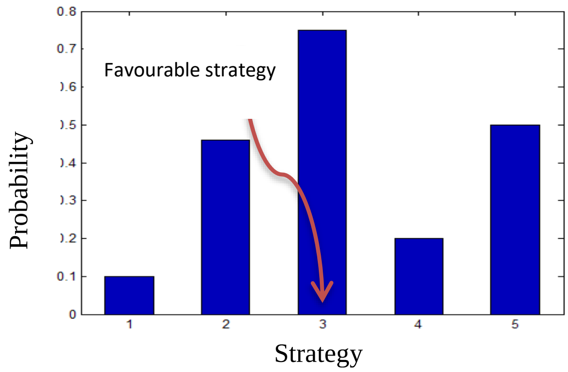
$\propto_{step}$ is constant which takes a value $\in (0,1]$and T is Boltzamann's temperature, which starts from $T \gg 0$ or $T = T_{intial}$ or $T \to \infty$, so K is a number of iterations and $s_i(q)$ is the Entropy Function of agent$i$. Clearly, each strategy has the maximum contribution for the minimisation of the expected utility function. This strategy is known as a favourable combined strategy $X_i^{[fav]}$ [9]. For example, there are 5 strategies for 3 agents as demonstrated in figure 3.



a) Favourable Strategy for Agent 1



b). Favourable Strategy for Agent 2



c) Favourable Strategy for Agent 3

Figure 3: Probability Distribution

All agents compute the objective function $(G(Y^{fav})^n$ where Y_{fav} is given by $Y^{fav} = \{X_1^{fav,n}, X_2^{fav,n}, .., X_{N-1}^{fav,n}, X_N^{fav,n}\}$. Actually, there are some criteria to terminate the algorithm of probability collectives either:

- If temperature $T \to 0$.
- If $\|\ G(Y^{fav})_n - G(Y^{fav})_{n-1} \| \leq \varepsilon$ where $\varepsilon > 0$.

For each iteration, the PC algorithm updates the boundaries of variables Ω and Boltzmann's temperature as follows:

$X_i^L(n+1) = (1-\lambda) * X_i^{fav}$, i=1… N (14)

$X_i^H(n+1) = (1+\lambda) * X_i^{fav}$, i = 1,………, N (15)

$T_{n+1} = (1-\alpha_T) * T_n$ (16)

Where $0 < \lambda < 1$ is the range factor and $0 < \alpha_T < 1$ is the cooling rate. The algorithm PC continues until one of mentioned criteria above is satisfied.

3. Application of PC to N-queens problem

In this thesis, we applied this problem addressed by PC to Distributed Constraint Satisfaction Problems (DCSP). We can formalise the algorithm as follows:

Step 1: Initialize the parameters of the PC algorithm $(T = 100, \lambda = 0.9, \alpha_s = 0.098, \alpha_T = 0.9, K = 150, M_i = 10, Runs = 20)$, and ($NQ = 40$ is a number of queens, $N = 10$ is a number of population and $NAction =$ is a number of actions.

Step 2: Create actions list.

Step 3: Initialize a set of agents where each agent can take a value from domain ={1,2,….,NQ} where initialization is done using a random permutation such as x=[1,3,4,2].
Step4: Apply actions and evaluate the objective function for each agent.
Step 5: Check all constraint which is satisfied.
Step 6: Find the best solution using the PC algorithm.
Step 7: Until the global minimum is reached, then accept the optimal solution.

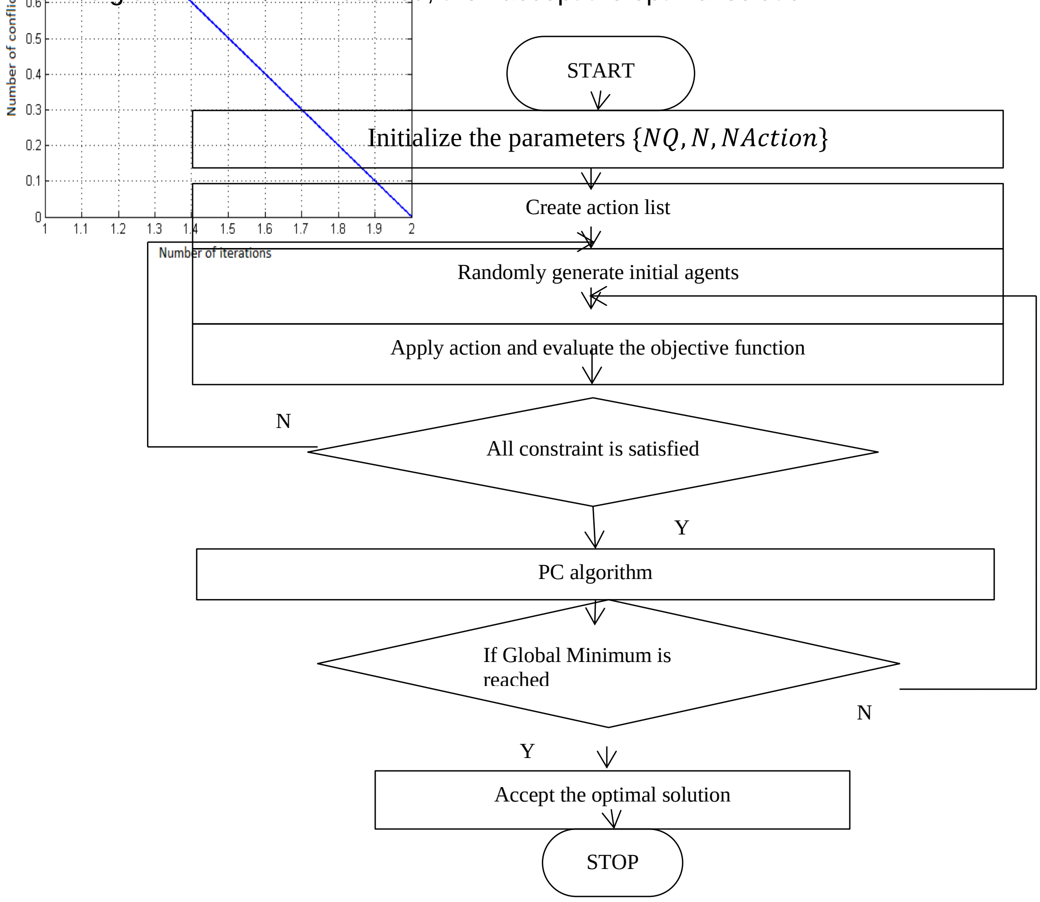
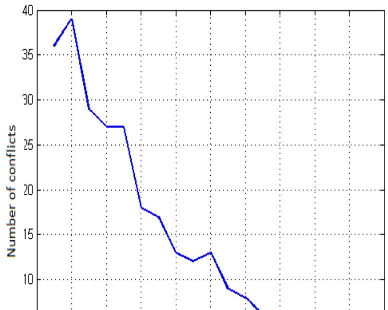


Figure 4: shows the flowchart of the PC algorithm to the N-queens problem.

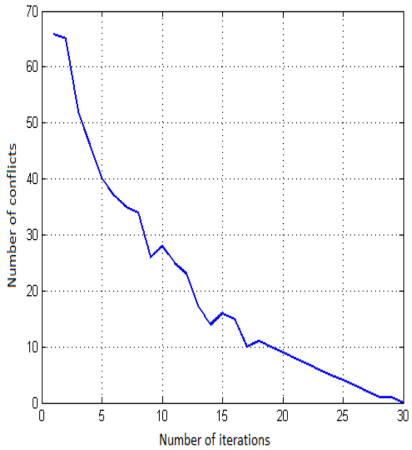
The probability collectives are implemented to solve four problems using Matlab2013b in Windows 7 operating systems and run on a personal computer with Intel Core(TM) i3 2.10 GHz CPU and 4.00GB of RAM. we implemented the PC algorithm of N-queens problem for various sizes N=(8,100,150,180) and show the best solution based on time and a number of iterations.

4. Results of N-Queens Problem

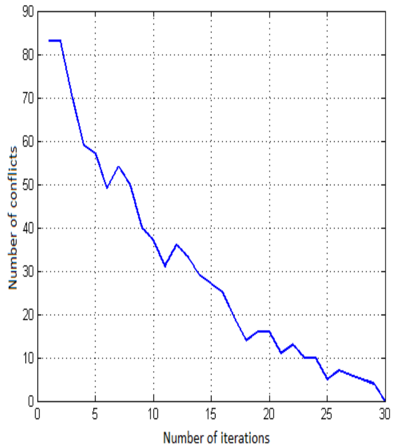
We implemented the PC algorithm to solve the N-queens problems at different sizes (N = 8,100,150,180). Figure 5 illustrates the convergence for various sizes of problem. The numbers of conflicts drop sharply from 1 to 0 as in figure 5 (a). Whereas, the number of conflicts in figure 5 (b, c, d) decreased gradually to reach zero. This means, the small size of the problem took much fewer iterations to convergences than larger sizes.



(a) N = 8 queens (b) N = 100 queens



(c) N = 150 queens



(d) N = 180 queens

Figure 5: The Convergence for PC Algorithm Runs on Different Sizes of Queens

Table 1 shows the result of applying the PC algorithm to find the best solution for N-queens, with the time that is needed to achieve each solution. For example, 8 queens are distributed on a chessboard as :(q_1 at column 4, q_1 at column 8, q_1 at column 1, q_1 at column 5, q_1 at column 7, q_1 at column 2, q_1 at column 6, q_1 at column 3).

Table 1: The Results of N-Queens Problem and Time of Reaching the Optimum Solutions

Number of queens	Number of Generations	Time	Solutions
8	2	0.408623 s	[4,8,1,5,7,2,6,3]
100	19	2168.79359 s	[27,90,17,62,47,2,41,71,24,3,80,4,37,66,96,87,29,69,28,65,67,1,54,57,84,94,40,44,89,95,11,73,31,34,19,38,98,100,45,76,8,52,46,14,53,36,77,42,81,78,56,5,10,79,18,43,13,93,97,55,83,51,15,63,85,70,33,91,6,99,68,21,64,48,82,20,25,23,7,39,92,59,12,22,49,26,88,60,32,75,72,74,61,35,50,30,58,16,9]

150	30	6067.60477 1s	[34,140,61,80,49,147,137,148,40,21,23,48,127,24 ,31,81,68,55,3,97,133,67,120,135,44,87,96,100, 33,89,2,25,14,116,130,9,5,136,11,142,129,94,45, 76,107,63,113,53,56,79,126,4,43,122,149,111,6, 57,20,26,46,42,144,131,118,17,146,65,77,10,134, 58,38,98,64,110,124,93,70,47,27,109,8,114,108, 95,105,138,117,132,60,22,19,1,71,145,13,90,86, 18,37,39,128,74,125,16,30,102,36,92,15,12,139, 73,75,78,112,62,115,28,121,35,52,7,99,32,91,50, 85,119,82,66,69,54,83,59,141,84,29,104,143,41, 123,88,51,101,72,106]
180	30	30240.0526 5s	[34,176,103,170,147,10,68,142,73,97,122,41,35, 172,155,65,109,61,168,171,123,137,62,134,100, 108,37,46,133,11,5,1,95,28,70,78,159,57,47,124, 148,143,39,24,135,4,48,52,140,71,119,141,113, 150,116,6,74,144,158,76,83,88,14,161,110,23,55, 96,69,44,20,84,127,120,85,111,25,131,51,12,87, 139,153,178,160,152,128,66,18,13,19,27,90,7,22, 145,164,180,17,45,54,101,59,129,174,177,16,126 ,72,166,36,3,33,82,99,26,175,21,156,179,107,43, 165,162,115,9,89,104,121,163,138,86,75,132,102 ,151,92,15,2,117,114,40,130,63,56,98,80,49,42, 93,154,118,112,58,94,64,146,53,105,125,91,30, 79,149,136,8,32,38,29,60,173,50,167,31,81,157, 77,169,67].

5. Conclusion

In this paper, we have implemented the N-queens problem using probability collectives algorithm. The PC is a general framework of agent coordination and distributed optimisation, which is a type of heuristic algorithm. This algorithm concentrates on adapting the distributions the strategy set of each agent to improve its performance. Each agent makes options using the determined utility until the algorithm reaches the convergence. The performance of probability collective is tested using the N-queens problems at different sizes ($N = 8,100,150,180$). The results show that the PC algorithm was successful and was sufficiently robust in solving these problems.

References

"Wikipedia," Wikimedia Foundation, 24 July 2016. [Online]. Available: HTTP:// en.wikipedia.org/wiki/Multi-agent_system.

Bieniawski, Stefan, D. H. Wolpert, and I. Kroo. "Discrete, continuous, and constrained optimisation using collectives." Proceedings of 10th AIAA/ISSMO Multidisciplinary Analysis and Optimisation Conference, Albany, New York. 2004.

Huang, Chien-Feng. "Combinatorial optimisation in biology using probability collectives multi-agent systems." Expert Systems with Applications 39.2 (2012): 1763-1771.

Kulkarni, Anand J., and Kang Tai. "Probability Collectives for Decentralised, Distributed Optimisation: a Collective Intelligence Approach." Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on. IEEE, 2008.

Kulkarni, Anand J., and Tai. "Probability Collectives: a Multi-agent Approach for Solving Combinatorial Optimisation Problems." Applied Soft Computing 10.3 (2010): 759-771.

Kulkarni, Anand Jayant, and Kang Tai. "Probability collectives: a distributed optimisation approach for constrained problems." IEEE Congress on Evolutionary Computation. IEEE, 2010.

Paul Cull, Rajeev Pandey, Isomphismand And The N-queens Problem, Technical report 94-20-02, 11 Feburary 1994.

Vasirani, Matteo, and Sascha Ossowski. "Collective-based multiagent coordination: a case study." International Workshop on Engineering Societies in the Agents World. Springer Berlin Heidelberg, 2007

Xu, Zixiang, Ahmet Unveren, and Adnan Acan. "Probability collectives hybridized with differential evolution for global optimisation." International Journal of Bio-Inspired Computation 8.3 (2016): 133-153.

Yang, Bo, and Ruiming Wu. "A modified probability collectives optimisation algorithm based on trust region method and a new temperature annealing schedule." Soft Computing 20.4 (2016): 1581-1600.

Zhao, Wei, and Nan Wang. "Probability Collectives using Response Surface estimation." Communications and Information Technology (ICCIT), 2013 Third International Conference on. IEEE, 2013.