


Parallel Big Bang–Big Crunch Global Optimization Algorithm: Performance and its Applications to routing in WMNs

Shakti Kumar¹  · Amar Singh¹ · Sukhbir Walia²

Published online: 13 April 2018

© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract This paper presents the performance of Parallel Big Bang–Big Crunch (PB3C) global optimization algorithm on CEC-2014 test suite. The performance is compared with 16 other algorithms. It has been observed that PB3C gave best performance on 7 functions of the test bench. Out of seven, for 6 functions it gave the unmatched best performance whereas on one count its performance was equaled by other algorithm as well. Further this paper proposes a PB3C based new routing approach to wireless mesh networks (WMNs). Being dynamic; routing is a challenging issue in WMNs. The approach is a near shortest path route evaluation approach. The approach was simulated on MATLAB. The performance was compared with 7 other approaches namely ad hoc on-demand distance vector, dynamic source routing, ant colony optimization, biogeography based optimization, firefly algorithm, BAT and simple Big Bang–Big Crunch based approaches. For WMNs of size 1000 nodes and above the PB3C was observed to outperform rest of the 7 algorithms.

Keywords PB3C · WMNs · AODV · DSR · ACO · BBO · FA · BAT

✉ Shakti Kumar
shaktik@gmail.com

Sukhbir Walia
drssaptu@gmail.com

¹ Computational Intelligence Laboratory, Baddi University of Emerging Sciences and Technology, Baddi, HP, India

² I.K. Gujral Punjab Technical University, Kapurthala, Jalandhar, Punjab, India

1 Introduction

The Big Bang Big Crunch (BB–BC) theory is one of the widely accepted theories of the evolution of our universe. Later this concept was formulated as a global optimization approach called BB–BC optimization algorithm [1–3]. Shakti et al. [4] extended it to a parallel BB–BC (PB3C) global optimization approach. PB3C is a multi-population approach and was demonstrated to have better convergence rate and accuracy than BB–BC based approach in a fuzzy model identification problem. The work reported here has been motivated by two objectives. The first one was to evaluate the PB3C performance on standards test functions and compare it with the performance of other algorithms. The second objective was to evaluate its performance on routing in WMNs. Routing in WMNs is a challenging issue that has attracted the attention of research community in the recent past [5]. Conventional static network shortest path routing approaches hit the road block in the case of WMNs where shortest path evaluation is quite difficult due to dynamic nature of WMNs in which almost all nodes can be mobile.

In order to assess the performance of PB3C algorithm and compare it with some of the commonly used algorithms available in the literature we evaluated the performance of PB3C algorithm on CEC-2014 test suite. It is an established fact today that in the case of complex problems falling in the category of NP hard or NP complete problems probability that a search will result into a best solution under a given time frame is very low and highly expensive computationally. Under such circumstances it has been observed that wherever best can be replaced with good enough solutions, soft computing approaches offer much better performance as compared to hard computing based approaches such as AODV, DSR etc. [6, 7]. Evaluation of shortest path under a given time constraint imposed due to node mobility in WMNs also belongs to this class of problems. Shakti et. al [8] presented 3 new integrated cost based, near shortest path routing approaches to wireless mesh networks. The algorithms performed reasonably well on small networks upto about 500 node WMNs. Their performance for higher node WMNs was observed to be unsatisfactory.

This paper evaluates the performance of PB3C algorithm on CEC 2014 benchmark and compares it with the performance of 16 other algorithms available in the literature. Further this paper proposes a PB3C based near shortest path routing approach to efficiently deal with the complex issue of routing in WMNs.

This paper is organized into 5 sections. Section 1 of the paper introduces the motivation for the paper, Sect. 2 discusses the simulation and performance of PB3C on CEC-2014 test bench and compares it with other algorithms. Section 3 proposes a PB3C algorithm based new routing approach, Sect. 4 discusses the implementation and performance of the approach for WMNs. The section compares the performance of proposed algorithm with 7 other algorithms found in the literature. Section 5 concludes the paper.

2 PB3C Algorithm and its Performance on CEC-2014 Test Bench

The original one population based big bang big crunch algorithm henceforth known as simple BBBC algorithm [1] has been modified to multi-population BB–BC called PB3C algorithm. The pseudo code for the PB3C approach is given as Algorithm 1. The Algorithm 1 and Algorithm 2 must update their elite(i) based upon local best of ith population. The algorithm was implemented in MATLAB and was tested using a Core i7 @ 2.2 GHz based laptop with 8 GB RAM. All the 30 functions of CEC-2014 test bench were tested with 25 trials for each function. We considered all functions with 10 dimensions

Table 1 Comparison of PB3C algorithm with other 16 algorithms on CEC-2014 benchmark functions

ALGORITHM	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
NRGA	2.790E+04	9.147E+02	1.517E+02	1.544E+01	1.961E+01	2.450E+00	2.030E-01	5.585E+00	8.694E+00	1.194E+02
FWA – DM	5.013E+03	1.342E-04	1.877E-09	1.413E+00	2.003E+01	7.063E-01	9.480E-02	2.536E-01	6.008E+00	1.593E+00
UMOEAS	0.000E+00	0.000E+00	0.000E+00	0.000E+00	1.683E+01	0.000E+00	0.000E+00	0.000E+00	2.725E+00	3.739E-01
SOO + BOBYQA	4.570E+03	3.600E-02	5.843E+03	0.000E+00	2.000E+01	2.000E-03	4.900E-02	1.890E+01	8.955E+00	1.304E+02
SOO	8.811E+06	6.343E+00	6.644E+00	6.780E-01	2.000E+01	2.000E-03	4.900E-02	1.890E+01	8.955E+00	1.304E+02
RSDE	0.000E+00	0.000E+00	0.000E+00	2.811E+00	1.922E+01	5.291E-02	3.550E-02	6.608E-01	8.522E+00	6.844E+01
POBL_ADE	1.620E+04	2.270E+03	5.740E-04	2.550E+01	1.910E+01	1.040E+00	1.630E-01	7.810E+00	7.630E+00	1.530E+02
FERDE	2.368E+00	6.288E-05	1.346E-03	0.000E+00	1.906E+01	8.890E-01	1.883E-02	0.000E+00	5.638E+00	3.674E-02
FCDE	0.000E+00	0.000E+00	0.000E+00	1.841E+01	2.033E+01	3.566E+00	1.961E-01	1.607E+01	2.099E+01	2.919E+02
DE_bbe6r1withrestart	0.000E+00	0.000E+00	0.000E+00	1.125E+00	1.845E+01	0.000E+00	1.688E-02	0.000E+00	4.895E+00	1.225E-03
CMLSP	1.769E-07	1.115E-15	1.056E-04	3.344E-15	1.686E+01	6.201E-02	0.000E+00	2.071E+00	1.659E+00	1.961E+02
GaAPADE	0.000E+00	0.000E+00	0.000E+00	3.069E+01	1.968E+01	1.484E-01	3.163E-03	0.000E+00	3.379E+00	1.518E-01
OptBees	7.842E+02	9.883E-03	9.213E-01	2.691E+00	2.000E+01	3.017E+00	1.562E-01	1.159E-13	2.084E+01	2.192E+02
LSHADE	0.000E+00	0.000E+00	0.000E+00	2.941E+01	1.415E+01	1.754E-02	3.043E-03	0.000E+00	2.345E+00	8.572E-03
RMA – LSCb – CMA	0.000E+00	0.000E+00	1.025E-07	8.501E-02	1.365E+01	1.479E-04	0.000E+00	0.000E+00	3.317E+00	7.678E+00
PB3C	4.660E+01	1.437E+01	6.268E-09	3.717E-04	1.644E-07	1.958E-02	6.642E-02	1.990E+00	1.791E+00	2.498E-01
MVMO	4.954E-04	7.098E-09	9.860E-11	9.546E+00	1.658E+01	3.445E-03	1.858E-02	6.687E-15	3.492E+00	2.137E+00

ALGORITHM	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20
NRGA	5.759E+02	1.242E-01	1.577E-01	2.537E-01	1.022E+00	2.747E+00	1.607E+04	7.420E+03	2.093E+00	1.719E+03
FWA – DM	3.722E+02	4.249E-02	1.206E-01	2.139E-01	7.748E-01	1.757E+00	2.545E+02	2.516E+01	1.299E+00	1.337E+01
UMOEAS	1.440E+02	0.000E+00	9.436E-03	1.100E-01	6.667E-01	1.530E+00	8.477E+00	7.840E-01	2.000E-01	3.706E-01
SOO + BOBYQA	3.491E+02	0.000E+00	3.000E+00	1.300E-01	4.200E-01	2.520E+00	4.226E+02	3.952E+03	5.500E-01	6.925E+03
SOO	3.491E+02	0.000E+00	3.000E+00	1.300E-01	4.400E-01	2.520E+00	3.123E+06	1.293E+04	5.500E-01	9.364E+03
RSDE	2.906E+02	2.206E-01	1.277E-01	1.360E-01	9.830E-01	2.233E+00	4.770E+01	1.996E+00	1.030E+00	7.215E-01
POBL_ADE	2.080E+02	2.690E-01	1.310E-01	2.600E-01	7.120E-01	1.410E+00	2.570E+02	3.320E+01	2.090E+00	1.260E+01

Table 1 continued

ALGORITHM	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20
<i>FERDE</i>	7.554E+01	1.227E-01	1.158E-01	9.359E-02	6.725E-01	1.530E+00	8.230E+00	2.730E+00	5.092E-01	1.704E+00
<i>FCDE</i>	6.605E+02	7.731E-01	3.556E-01	3.470E-01	1.594E+00	3.191E+00	3.028E+02	2.271E+01	2.463E+00	1.778E+01
<i>DE_bbe6r1withrestart</i>	1.965E+02	2.929E-01	1.281E-01	1.113E-01	8.317E-01	1.872E+00	1.398E+00	6.207E-01	1.418E-01	5.593E-02
<i>CMLSP</i>	1.530E+02	3.027E-02	2.725E-02	1.892E-01	8.966E-01	1.555E+00	3.127E+02	3.085E+01	1.251E+00	1.994E+01
<i>GaAPADE</i>	1.831E+02	1.402E-01	6.009E-02	9.424E-02	6.057E-01	1.977E+00	9.914E+00	2.230E-01	2.566E-01	4.316E-01
<i>OptBees</i>	3.927E+02	1.304E-01	4.162E-01	3.687E-01	2.439E+00	2.640E+00	6.844E+02	3.350E+01	9.330E-01	8.958E+00
<i>LSHADE</i>	3.206E+01	6.817E-02	5.156E-02	8.136E-02	3.661E-01	1.241E+00	9.767E-01	2.441E-01	7.730E-02	1.849E-01
<i>RMA - LSCb - CMA</i>	2.013E+01	1.646E-02	3.292E-02	1.265E-01	4.715E-01	1.054E+00	7.834E+01	5.221E+00	7.661E-02	8.057E+00
<i>PB3C</i>	2.982E+00	8.414E-03	1.685E-02	5.681E-04	3.315E-01	1.932E-01	2.433E+01	2.520E+00	2.752E-01	1.517E+00
<i>MVMO</i>	9.628E+01	4.223E-02	3.553E-02	8.906E-02	4.346E-01	1.449E+00	9.357E+00	7.826E-01	1.583E-01	3.126E-01
ALGORITHM	F21	F22	F23	F24	F25	F26	F27	F28	F29	F30
<i>NRGA</i>	4823.4272	37.56658082	329.4574872	130.7641476	183.6782269	100.13662	280.77757	477.14738	413.29099	1727.5377
<i>FWA - DM</i>	9.464E+01	3.409E+01	3.295E+02	1.274E+02	1.787E+02	1.001E+02	3.213E+02	3.472E+02	2.117E+02	3.943E+02
<i>UMOEAS</i>	5.404E-01	2.448E-01	3.295E+02	1.083E+02	1.260E+02	1.000E+02	2.548E+01	3.129E+02	1.955E+02	2.339E+02
<i>SOO + BOBYQA</i>	1.940E+03	1.265E+02	2.000E+02	1.157E+02	1.391E+02	1.001E+02	2.000E+02	2.000E+02	2.000E+02	2.000E+02
<i>SOO</i>	2.469E+04	1.265E+02	2.000E+02	1.157E+02	1.452E+02	1.001E+02	2.000E+02	2.000E+02	2.000E+02	2.000E+02
<i>RSDE</i>	1.209E+00	1.165E+01	3.295E+02	1.191E+02	1.295E+02	1.001E+02	9.125E+01	3.869E+02	2.126E+02	5.052E+02
<i>POBL_ADE</i>	1.030E+02	3.000E+01	3.290E+02	1.240E+02	1.860E+02	1.000E+02	2.560E+02	4.230E+02	3.550E+05	6.380E+02
<i>FERDE</i>	8.543E+00	3.242E+00	3.295E+02	1.146E+02	1.363E+02	1.001E+02	3.664E+02	3.664E+02	3.182E+02	5.348E+02
<i>FCDE</i>	1.481E+02	2.750E+01	3.295E+02	1.369E+02	1.840E+02	1.003E+02	4.752E+01	4.569E+02	3.405E+04	8.667E+02
<i>DE_bbe6r1withrestart</i>	7.867E-01	1.541E-01	3.295E+02	1.122E+02	1.209E+02	1.001E+02	6.161E+01	3.634E+02	2.178E+02	4.673E+02
<i>CMLSP</i>	3.639E+01	8.953E+01	2.018E+02	1.099E+02	1.275E+02	1.000E+02	4.113E+01	2.803E+02	2.000E+02	2.164E+02
<i>GaAPADE</i>	5.086E-01	3.247E+00	3.295E+02	1.089E+02	1.636E+02	1.001E+02	8.969E+01	3.832E+02	2.223E+02	4.672E+02
<i>OptBees</i>	5.706E+01	1.702E+01	2.724E+02	1.374E+02	1.460E+02	1.004E+02	7.423E+00	3.067E+02	2.200E+02	3.892E+02

Table 1 continued

ALGORITHM	F21	F22	F23	F24	F25	F26	F27	F28	F29	F30
LSHADE	4.081E-01	4.410E-02	3.295E+02	1.075E+02	1.327E+02	1.001E+02	5.806E+01	3.808E+02	2.220E+02	4.649E+02
RMA – LSCb – CMA	4.929E+01	8.475E+00	3.295E+02	1.084E+02	1.751E+02	1.000E+02	1.848E+02	3.887E+02	2.271E+02	5.851E+02
PB3C	3.316E+00	1.763E+00	2.745E+02	1.060E+02	1.012E+02	1.000E+02	1.536E-01	3.568E+02	2.281E+02	4.971E+02
MVMO	1.935E+00	2.629E-01	3.295E+02	1.092E+02	1.161E+02	1.000E+02	1.720E+01	3.611E+02	1.814E+02	4.917E+02

Table 2 Number of functions for which PB3C gave the best performance

Functions for which PB3C gave best performance	f11, f14, f15, f16, f24, f27
Function for which PB3C gave best performance that was equaled by other algorithms also.	f26

Table 3 Comparative performance of various algorithms on CEC-2014 test bench

Algorithm name	No. of functions for which this algorithm performed unequalled best	No. of functions for which best performance is observed but is equaled by other algorithms also	Total no. of functions in which best performance was observed
SOO+BOBYQA	0	4	4
DE_b6e6rlwithrestart	0	5	5
FCDE	0	3	3
FERDE	0	2	2
UMOEAS	0	9	9
CMLSP	0	3	3
RMALSchCMA	0	3	3
LSHADE	3	4	7
NRGA	0	0	0
FWADM	0	1	1
GaAPADE	0	4	4
POBL_ADE	0	1	1
MVMO	1	2	3
OptBees	0	1	1
RSDE	0	3	3
SOO	0	3	3
PB3C	6	1	7

only. The mean value of all the 30 functions is placed in Table 1. The functions on which PB3C gave unmatched results are given in Table 2.

The Table 3 indicates that PB3C gave best performance with minimum error for 7 functions. Out of these 7 functions there are 6 such function for which it gave performance unmatched by any other algorithm. For test function 26 it achieved the best results which was also achieved by 5 other algorithms namely UMOEAS, POBL_ADE, CMLSP, RMALSchCMA and MVMO. When we compare LSHADE and PB3C we find that both give minimum mean error on 7 test functions. However, LSHADE gives best performance on 3 test functions which no other algorithm could achieve and best performance (minimum mean error) on 4 other test functions namely function number f1, f2, f3 and f8; this best performance was also achieved by other algorithms namely FCDE, UMOEAS, DE_b6e6rlwithrestart, GaAPADE, RMALSchCMA and RSDE as well. As far as comparison with UMOEAS is concerned; UMOEAS performs best on 9 test functions but the UMOEAS is not unique on this count. This best performance of UMOEAS is also matched

by other algorithms namely FCDE, DE_b6e6rlwithrestart, GaAPADE, RMALSchCMA, RSDE and SOO as well. Thus the observations clearly show that PB3C delivers unmatched best performance in 6 functions which no other algorithm could reach.

Algorithm 1 Parallel Big Bang Big Crunch (PB3C) Algorithm

```

Begin
/* Big Bang Phase */
Generate N populations each of size NC candidates randomly Every candidate consisting
of NG genes;
iter = 1
/* End of Big Bang Phase */
while iter <= TC do
/* TC is a termination criterion */
for i = 1: N do
/* Big Crunch Phase Starts */
Compute the fitness of each candidate solutions;
Sort the population from best to worst based on fitness (cost) value;
Compute the centre of mass using Equation (1) of  $i^{th}$  population [1];

```

$$x^{\rightarrow c} = \frac{\sum_{j=1}^{NC} \left[\frac{1}{f_j} x^{\rightarrow j} \right]}{\sum_{j=1}^N \frac{1}{f_j}} \quad (1)$$

```

Best fit individual can be chosen in place of centre of mass;
Select local best candidates  $\ell_{best(i)}$  for  $i^{th}$  population;
end for
From amongst N best candidates select the globally best gbest candidate;
for i = 1: N do
With a given probability replace a gene of  $\ell_{best(i)}$  with the corresponding gene of
global best (gbest) candidate;
end for
/* End of Big Crunch Phase */
/* Big Bang Phase Starts */
Each of the N Current generations(populations) has one centre of mass/elite.
Evolve a new generation around every elite as follows:
for j = 1: NC do
for k = 1: NG do

```

```

Calculate new candidate solutions around the centre of mass/elite by adding
or subtracting a normal random number whose value decreases as the iterations elapse
using Equation 2;

```

$$x_{new(j,k)} = \ell_{best(j,k)} + (L * rand) / m \quad (2)$$

```

end for
end for
/* m is the number of iteration and L is upper limit of the parameter */
/* xnew is the new candidate solution */
/* End of Big Bang Phase */
iter = iter + 1;
end while
End

```

3 PB3C Algorithm Based Routing Approach to WMNs

Wireless Mesh Networks (WMNs) have drawn significant attention of researchers in the recent past. These are rapidly deployable, self-aware, self-organizing, self-configured, self-healing and self-balancing networks. In WMNs within a given radio range each node (stationary or mobile) has the capability to join and create a network automatically. It is able to do so by sensing neighboring nodes with a similar capability within their radio range and admitting themselves to the radio network according to the given set of rules. Routing in WMNs is the process to direct data packets from a given source to a given destination. It requires significant attention in the dynamic network conditions of a WMN. It is desired that these routing algorithms must work in a decentralized self-organizing and self-configuring way in a highly dynamic environment. Due to their dynamic nature, routing in WMNs is a highly complex issue. Increasing size of WMNs further compounds the routing challenge. Due to complexities associated with exact reasoning there is an increasing demand for soft computing based techniques in WMN research. These techniques may make the WMNs more popular in terms of their self-organizing and self-configuring capabilities. Soft computing provides the optimal solution within an affordable resource/time permitted by the WMN dynamics. Since, computation times are very limited, the best solution needs to be replaced by good enough solutions such that a given WMN quickly adapts to dynamically changing environment [9]. The Parallel Big Bang and Big Crunch (PB3C) optimization is one such powerful tool that can help simplify the things. PB3C approach can be used to select the near shortest path for appropriate route selection in WMNs. The underlining principle is to compute appropriate path within a stipulated time frame and use it rather than the shortest path that may not get computed within a given time frame.

This appropriate path is the optimal cost path that may or may not be a least cost path. We shall call this optimal cost path as the 'near shortest path'. There are many different routing metrics available in the literature. Some of these are minimum hop count, per hop Round Trip Time (RTT) [10], Per-Hop Packet Pair Delay (PktPair) [11], Expected Transmission Count (ETX) [12], Expected Transmission Time (ETT), Weighted Cumulative ETT (WCETT) [13], Expected Transmission on a Path (ETOP) [14], Effective Number of Transmission (ENT) and Modified Expected Number of Transmissions (mETX) [15], Metric of Interference and Channel Switching (MIC) [16], Bottleneck Link Capacity (BLC) path metric [17]. A novel interference aware low overhead routing metric was proposed by Liran Ma et al. [18], cross layer link quality and congestion aware (LQCA)[19]. For the WMNs integrated link cost route function (ILC) was defined as follows [6, 7, 20]:

$$\text{integrated link cost (ILC)} = f(\text{throughput, delay, jitter, node_residual}_{energy})$$

Further we use the same route cost model for evaluating the cost as used in [20]

As discussed earlier PB3C is a multi-population based algorithm. In the PB3C algorithm based approach we generate a set of N populations of candidate solutions. Each population consists of NC candidate solutions. A route between source terminal pair is one potential candidate that may lead to near shortest path between source and terminal node. For our model we consider WMN as a connected, directed topology graph with nodes and links between adjacent nodes with respective ILCs. Our objective is to evolve an ILC based shortest possible path within a given timing constraint. We assume WMN under consideration to be static for the given time period (i.e. time constraint). We effect all movements of nodes after given timing constraint. For the next timing slot (time constraint) we consider network to be again static and recomputed the shortest path within the available time.

In order to create initial candidate solutions we construct the initial population as follows [9]:

We start to search a random path from source to destination by randomly selecting a node from the set of neighboring node of source node. Then, a node from the neighborhood of this node is selected randomly. We continue with this process until the destination node is reached. In order to keep the paths loop-free, those nodes which are already included in the current path are excluded from being selected as the next node to be added into the path, thereby avoiding re-entry of the same node into a path. This gives us a path between source and the destination node. Repeating this process, we get the initial population of N paths. The path with the lowest cost is selected as the elite.

In order to generate next generation from elite, we use the following steps to evolve a new candidate solution (route) from the elite:

- (a) Generate a random number R between 1 and S , Where S is the size of the route/path. It is the total number of nodes in the elite ($\ell_{best(i)}$).
- (b) Retain first R nodes of the elite discarding the remaining path.
- (c) From R th node onward discover a path to destination node. Thus evolving a new path between source and terminal node.

We continue the above procedure till the next generation with NC routes (Candidate Solutions) has evolved. We generate N such generations/populations.

Algorithm 2 PB3C Algorithm for Dynamic Shortest Path Routing

```

Begin
/* Big Bang Phase Starts */
Initialize the PB3C parameters. Generate  $N$  populations each with randomly generated
NC candidate Solutions (Paths);
/* End of Big Bang Phase */
while  $\neq TC$  do
/* TC is a termination criterion */
for  $i = 1: N$  do
/* Big Crunch Phase Starts */
Calculate ILC of every link of the network. Using ILC evaluate the fitness/cost of
every discovered path;
Sort the  $i$ th populations from best to worst based on the values of ILC;
The best fit individual of  $i$ th population is chosen as the local best of the particular
population;
end for
/* End of Big Crunch Phase */
/* Big Bang Phase Starts */
Record the global best (gbest) path from amongst all the local best paths;
for  $i = 1: N$  do
Generate a random number  $R$  between 1 and  $S$ , Where  $S$  is the total number total
number of nodes in the elite ( $\ell_{best(i)}$ );
Retain first  $R$  nodes of the elite discarding the remaining path;
From  $R^{th}$  node onward discover a path to destination node. Thus evolving a new
path between source and terminal node;
end for
/* End of Big Bang Phase */
end while
Print  $G_{best}$ 
End

```

Table 4 Architectural details of various client WMN scenarios

No. of nodes	Area (mm)	Radio range	Timing constraint(in s)
100	500 × 500	250	0.1, 0.2, 0.4, 0.6, 0.8, 1, 1.2, 1.4, 1.6, 1.8, 2.0
500	1000 × 1000	250	0.1, 1.2, 1.4, 1.6, 1.8, 2, 2.2, 2.4, 2.6, 2.8, 3.0
1000	1000 × 1000	250	1.4, 1.6, 1.8, 2, 2.2, 2.4, 2.6, 2.8, 3, 3.2, 3.5
1500	2000 × 2000	250	1.4, 1.6, 1.8, 2, 2.2, 2.4, 2.6, 2.8, 3, 3.5, 4.0
2000	2000 × 2000	250	1.5, 2, 2.2, 2.4, 2.6, 2.8, 3, 3.5, 4, 4.5, 5.0
2500	2000 × 2000	250	2, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.0, 7.5, 8.0, 8.5, 9.0, 9.5, 10.0

4 Performance of the Proposed Routing Approach: Simulation and Results

We implemented the proposed approach in MATLAB and performed simulations considering dynamic scenarios. The architectural details of various Client WMN scenarios are given in Table 4. We considered 100, 500, 1000, 1500, 2000 and 2500 node client WMN architectures. The proposed PB3C based routing approach was applied to all these networks. Performance of all these networks was evaluated for stipulated time constraints. For a given network and for each timing constraint we conducted the 20 trials for each set. For each time set minimum, average and maximum cost paths were enumerated. In this case the stop and re-evaluate criterion was defined as the available computing time in the form of given time constraint. Though we performed observations for 100, 500, 1000, 2000 and 2500 node WMNs, for the sake of brevity we have focused our discussion on the performance of large networks consisting of 1500, 2000 and 2500 node client WMNs only. We have included the performance of small networks consisting of 100, 500 and 1000 nodes towards the end of this section.

4.1 Comparative Performance of 1500 Node Client WMNs

For 1500 node client WMNs we evaluated the performance of eight algorithms. We conducted 11 sets of trials; each set consisted of 20 trials. The results are placed as Table 5 and are also shown as histogram in Fig. 1. We observe that for a processing time constraint of 1.4 s AODV, ACO and DSR failed to generate a path in any of the 20 trials of the set. PB3C topped the table by generating the minimal cost path in 10+A trials out of the total number of 20 trials of the set. PB3C produced minimal cost path 10+A = 18 times. Here, by 10+A = 18 means in 10 trials algorithm produced minimum cost path from amongst all the algorithms and A= 8 is the number of trials for which BBO, BBBC, Firefly (FA), PB3C and BAT algorithms produced the same result in this particular set only. 2 times PB3C generated paths costlier than minimum cost paths generated by other algorithms. BBBC generated minimal cost path 1+A = 9 times, FA generated 1+A = 9 times.

We further, observe that except for timing constraints of 2.6, 3.5 and 4 s, PB3C outperformed all other 7 competitors. For the timing constraint of 2.6 s BBBC evaluated the minimum cost route 6+G = 11 times followed by PB3C 5+G = 10 times and FA 4+G = 9 times. G = 5 is the number of trials for which BBO, BBC, FA, PB3C and BAT algorithms

Table 5 Performance of 1500 node client network

Algorithm	Timing constraints										
	1.4	1.6	1.8	2.0	2.2	2.4	2.6	2.8	3.0	3.5	4
AODV	–	–	–	–	–	–	–	–	–	–	5+15 FAIL
DSR	–	–	–	–	–	–	–	–	–	–	–
ACO	–	–	–	–	–	–	–	–	–	–	–
BBO	A	B	C	D	E	F	G	H	I	J	K
BBBC	1+A	3+B	1+C	3+D	8+E	6+F	6+G	7+H	6+I	9+J	5+K
FIREFLY	1+A	B	1+C	3+D	2+E	3+F	4+G	3+H	3+I	J	4+K
PB3C	10+A	11+B	14+C	11+D	8+E	8+F	5+G	9+H	9+I	8+J	5+K
BAT	A	B	C	D	E	F	G	H	I	J	K

A = 8, B = 6, C = 4, D = 3, E = 2, F = 3, G = 5, H = 1, I = 2, J = 3, K = 1, – means failed to produce path in any of the trials

Best Performance Frequency Vs Time Plot (Number of Nodes : 1500, Area : 2000m×2000m)

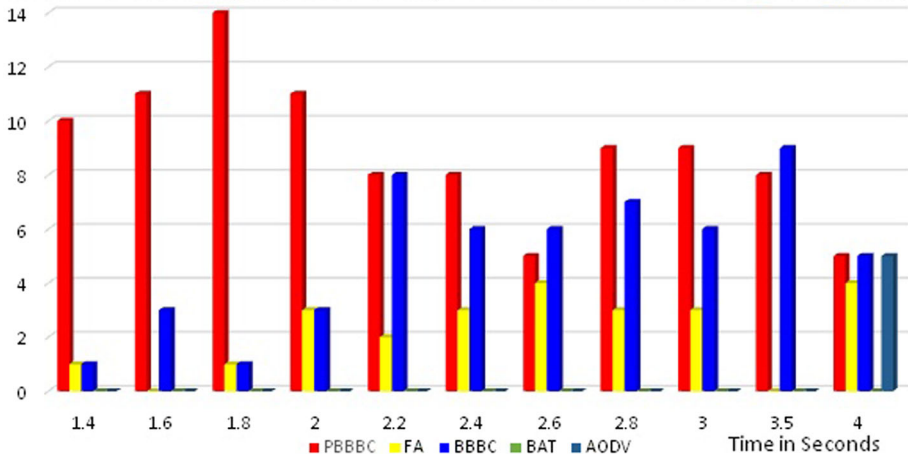


Fig. 1 Comparative performance OF 1500 node client WMNs

gave equal performance i.e. produced minimum cost paths with equal ILC. For the timing constraint of 3.5 s BBBC evaluated the minimum cost route 9+J = 12 times followed by PB3C 8+J = 11 times; J = 3 is the number of trials for which BBO, BBC, FA, PB3C and BAT algorithms gave equal cost paths. ACO, DSR and AODV failed to discover route in any of the trial sets. At the timing constraint of 4 s PB3C and BBBC generated minimum cost path 5+K = 6 times followed by FA 4+K = 5 times. Though, AODV could discover minimum cost route 5 times out of 20 trials yet it cannot be recommended for use in client WMNs due to the fact that it failed to generate any path in 15 of the 20 trials. Indicating that AODV is highly unreliable for routing in client WMNs.

As shown in Table 6 (Comparative Performance of all algorithms) for the total of 220 trials from 11 sets PB3C gave the best performance 98 times followed by BBBC 55 times, FA 24 times and AODV 5 times. 38 times BBO, BBC, FA, PB3C and BAT algorithms achieved same performance except for ACO, AODV and DSR.

Table 6 Comparative performance of all algorithms

Timing constraints										
Number of nodes	Trials	PB3C	FA	BBBC	BAT	AODV	BBO	DSR	ACO	All equals
100	220	74	17	75	0	35	4	0	0	15
500	220	75	16	76	0	34	0	1	0	18
1000	220	98	37	72	0	0	0	0	0	13
1500	220	98	24	55	0	5	0	0	0	38
2000	220	106	21	51	0	0	0	0	0	42
2500	340	167	32	91	0	6	0	0	0	44

Table 7 Performance OF 2000 Node Client Network

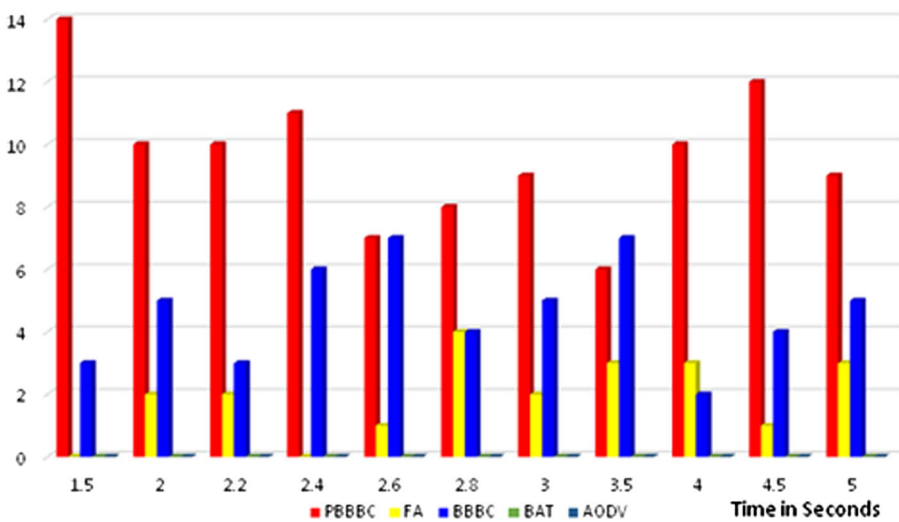
Algorithm	Timing constraints										
	1.5	2.0	2.2	2.4	2.6	2.8	3.0	3.5	4.0	4.5	5
AODV	-	-	-	-	-	-	-	-	-	-	-
DSR	-	-	-	-	-	-	-	-	-	-	-
ACO	-	-	-	-	-	-	-	-	-	-	-
BBO	A	B	C	D	E	F	G	H	I	J	K
BBBC	3+A	5+B	3+C	6+D	7+E	4+F	5+G	7+H	2+I	4+J	5+K
FA	A	2+B	2+C	D	1+E	4+F	2+G	3+H	3+I	1+J	3+K
PB3C	14+A	10+B	10+C	11+D	7+E	8+F	9+G	6+H	10+I	12+J	9+K
BAT	A	B	C	D	E	F	G	H	I	J	K

A = 3, B = 3, C = 5, D = 3, E = 5, F = 4, G = 4, H = 4, I = 5, J = 3, K = 3, - means failed to produce path in any of the trials

4.2 Comparative Performance of 2000 Node Client WMNs

Table 7 and Fig. 2 present simulation results for 2000 node client WMNs. We observe that except for timing constraints of 2.6 and 3.5 s, PB3C outperformed all other competitors. At timing constraint of 2.6 s PB3C and BBBC gave equal performance by generating minimum cost paths 7+E = 12 times each, followed by FA 1+E = 6 times, In E = 5 trials out of 20, PB3C, BBO, FA, BAT and BBBC gave equal performance. At timing constraint of 3.5 s BBBC evaluated the minimum cost route 7+H = 11 times followed by PB3C, 6+H = 10 times and FA, 3+H = 7 times. ACO, AODV and DSR failed to discover a route in any of the trials of any of the sets.

As shown in Table 6 we find that for the total of 220 trials from 11 sets PB3C gave the best performance 106 times followed BBBC 51 times, FA 21. In 42 trials 5 algorithms namely BBBC, BBO, BAT, FA and PB3C achieved same performance.

Best Performance Frequency Vs Time Plot (Number of Nodes : 2000, Area : 2000mx2000m)**Fig. 2** Comparative performance of 2000 node client WMNs

4.3 Comparative Performance of 2500 Node Client WMNs

For 2500 nodes networks 17 sets each consisting of 20 trials were conducted. Table 8 and histogram of Fig. 3 present the simulation results.

We observe that for all the timing constraints PB3C outperformed all other 7 algorithms. Up to time constraint of 8.5 s AODV failed to discover route in any of the trial sets. With the timing constraint going up to 9 s AODV discovered the route 4 times and failed to discover the route 16 times out of the total of 20 trials for this constraint. For both timing constraints of 9.5 and 10 s AODV succeeded to discover rout only once in each set and failed 19 times. Thus for the given client networks the AODV is highly unreliable protocol. Further the ACO and DSR could not discover the route in any of the trial sets.

For the total of 340 trials from 17 sets PB3C pulled up the best performance 167 times followed BBBC 91 times, FA 32 times, AODV 6 times and 44 times 5 algorithms i.e. BBBC, BBO, BAT, FA and PB3C based algorithms achieved same performance.

4.4 Comparative Performance of All Approaches

For comparative performance of the given 8 WMN routing approaches we have conducted total of 1440 trials. We have included the observations on WMNs with 100, 500 and 100 nodes also here. As is evident in Fig. 4, the performance of BBBC and PB3C is better than other approaches for 100 and 500 node client WMNs. But when the number of nodes increases beyond 1000 nodes the performance of PB3C showed significant improvement, out performing all other competing approaches. Hence, for complex and large networks of 1000 node capacity and beyond the PB3C approach is observed to be the best suited approach as compared to given other soft computing and hard computing based approaches.

Table 8 Performance of 2500 node client network

Algo	Timing constraints (in s)																		
	2	2.5	3	3.5	4	4.5	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10		
AODV	-	-	-	-	-	-	-	-	-	-	-	-	-	-	4+16 FAIL	1+19 FAIL	1+19 FAIL		
DSR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
ACO	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
BBO	A	B	C	D	E	F	G	H	0	I	J	K	L	M	N	O	P		
BBBC	3	1	5	5	6	7	6	6	5	7	5	8	6	6	4	6	5		
FA	5	5	1	0	4	0	2	2	4	0	2	0	1	0	2	2	2		
PB3C	8	11	12	12	7	9	10	10	11	12	10	10	10	10	7	8	10		
BAT	A	B	C	D	E	F	G	H	0	I	J	K	L	M	N	O	P		

A = 4, B = 3, C = 2, D = 3, E = 3, F = 4, G = 2, H = 2, I = 1, J = 3, K = 2, L = 3, M = 4, N = 3, O = 3, P = 2, - means failed to produce path in any of the trials

Best Performance Frequency Vs Time Plot (Number of Nodes : 2500, Area : 2000mx2000m)

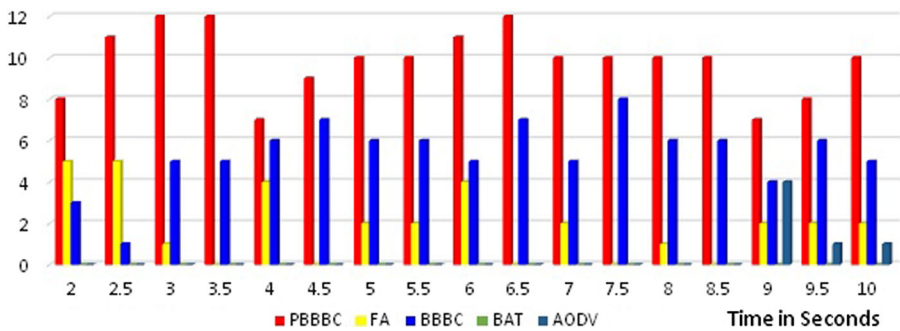


Fig. 3 Comparative performance of 2500 node client WMNs

Best Performance Frequency Vs Number of Nodes (Total number of trials = 1440)

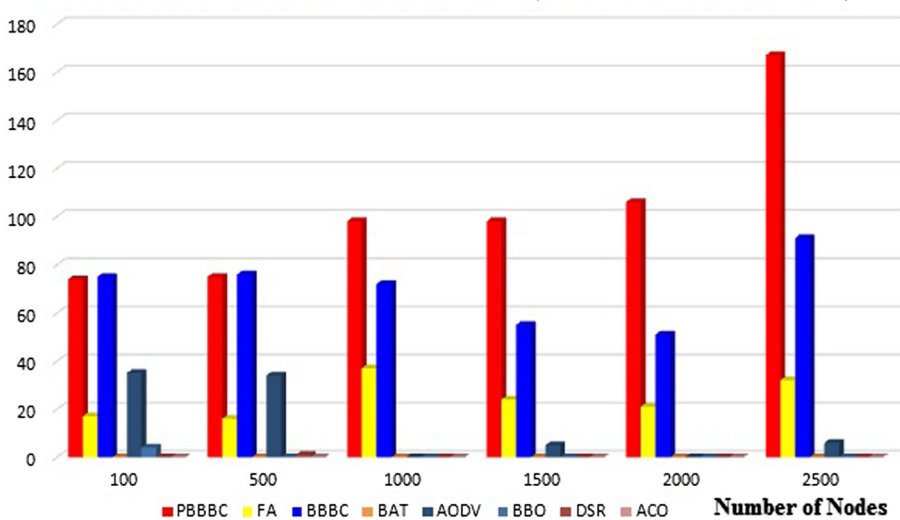


Fig. 4 Comparative performance of all algorithm

5 Conclusions

This paper evaluated the performance of PB3C algorithm on CEC-2014 test bench and compared it with 16 other commonly used algorithms found in the literature. Further PB3C algorithm was applied to near shortest path routing in WMNs. Routing under dynamic conditions as is the case with WMNs is a challenging issue faced by research community. As far as performance of PB3C over CEC-2014 test bench is concerned it gave best performance in 7 test bench functions. Out of these 7 it achieved unmatched best for 6 functions. Seventh best performance was achieved for test function number 26 of CEC-2014. However, this performance was also equalled by 5 other algorithms namely UM0EAS, POBL_ADE, CMLSP, RMA1SChCMA and MVMO. Though UM0EAS gave minimal error on 9 functions yet this performance was equaled by other algorithms as well. Thus we find PB3C is the only algorithm that gives unique unmatched best performance in

6 of the test functions making it the best available algorithms out of the 17 algorithms. A PB3C based new routing approach was also proposed to evaluate near shortest path based routing in client WMNs. The approach was compared with the 7 other approaches found in the literature. Simulation results prove that the PB3C based routing approach outperforms all other 7 approaches when the network size grows to 1000 nodes or above. Thus establishing the clear superiority of the proposed approach over AODV, DSR, ACO, FA, BAT, BBO and BB-BC algorithm based approaches.

References

- Erol, K., & Eksin, I. (2006). A new optimization method: Big Bang–Big crunch. *Advances in Engineering Software*, 37, 106–111.
- Kumbasar, T., Yesil, E., Eksin, I., & Guzelkaya, M. (2008). Inverse fuzzy model control with online adaptation via Big Bang–Big crunch optimization. In *ISCCSP 2008*, Malta (pp. 697–703) (March 12–14, 2008).
- Kripka, M., & Kripka, R.M.L. (2008). Big crunch optimization method. In *International conference on engineering optimization*, Brazil.
- Kumar, S., Walia, S. S., & Singh, A. (2013). Parallel Big Bang–Big Crunch algorithm. *International Journal of Advanced Computing, Recent Science Publications*, 46(3), 1330–1335.
- Waharte, S., Boutaba, R., Iraqi, Y., & Ishibashi, B. (2006). Routing protocols in wireless mesh networks: Challenges and design considerations. *Multimedia Tools Applications*, 29(3), 285–303.
- Kumar, S., Singh, B., & Sharma, S. (2013). Soft computing framework for routing in wireless mesh networks: An integrated cost function approach. *International Journal of Electronics Computer and Communications Technologies (IJECCCT)*, 3(3), 25–32.
- Sharma, S., Kumar, S., & Singh, B. (2014). Hybrid intelligent routing in wireless mesh networks: Soft computing based approaches. *International Journal of Intelligent Systems and Applications, Modern Education and Science Press*, 1, 45–57.
- Kumar, S., Singh, B., & Sharma, S. (2015). Routing in wireless mesh networks: Three new nature inspired approaches. *Wireless Personal Communications: An International Journal*, 83(4), 3157–3179.
- Yang, S., Cheng, H., & Wang, F. (2010). Genetic algorithms with immigrants and memory schemes for dynamic shortest path routing problems in mobile ad hoc networks. *IEEE Transactions on Systems, MAN, and Cybernetics-Part C: Applications and Reviews*, 40(1), 52–63.
- Adya, A., Bahl, P., Padhye, J., Wolman, A., & Zhou, L. (2004). A multi radio unification protocol for IEEE 802.11 wireless networks. In *International conference on broadcast networks (Broad Nets)*, San Jose, California, USA, (pp. 344–354) (October 25–29, 2004).
- Yang, Y., Wang, J., & Kravets, R. (2005). Interference-aware load balancing for multihop wireless networks. In *Technical Report UIUCDCS-R-2005-2526, Department of Computer Science*, University of Illinois at Urbana-Champaign (pp. 1–16).
- De Couto, D.S.J., Aguayo, D., Bicket, J., & Morris, R. (2003). A high-throughput path metric for multihop wireless routing. In *Proceedings of ACM annual international conference on mobile computing and networking (MOBICOM)*, San Diego, CA, USA (pp. 134–146) (September 14–19, 2003).
- Draves, R., Padhye, J., & Zill, B. (2004). Comparisons of routing metrics for static multi-hop wireless networks. In *ACM annual conference of the special interest group on data communication (SIGCOMM)*, Portland, Oregon, USA (pp. 133–144) (August 30–September 03, August 2004).
- Jakllari, G., Eidenbenz, S., Hengartner, N., Krishnamurthy, S., & Faloutsos, M. (2008). Link positions matter: A non commutative routing metric for wireless mesh networks. In *Proceedings of IEEE annual conference on computer communications (INFOCOM)*, Phoenix, Arizona, USA (pp. 744–752) (April 13–18, 2008).
- Koksal, C. E., & Balakrishnan, H. (2006). Quality-aware routing metrics for time-varying wireless mesh networks. *IEEE Journal on Selected Areas in Communications*, 24(11), 1984–1994.
- [Draves, R., Padhye, J., & Zill, B. (2004). Routing in multi-radio. Multihop wireless Mesh networks. In *ACM annual international conference on mobile computing and networking (Mobi Con04)*, Philadelphia, Pennsylvania, USA (pp. 114–128).
- Liu, T., & Liao, W. (2006). Capacity-aware routing in multi-channel multi-rate wireless mesh networks. In *Proceedings of IEEE international conference on communications (ICC)* (pp. 1971–1976).

18. Ma, L., Zhang, Q., Xiong, Y., & Zhu, W. (2005). Interference aware metric for dense multi-hop wireless network. In *Proceedings of IEEE international conference on communications (ICC)* (pp. 1261–1265).
19. Karbaschi, G., & Fladenmuller, A. (2005). A link quality and congestion-aware cross layer metric for multi-hop wireless routing. In *Proceedings of IEEE MASS05* (pp. 7–11).
20. Kumar, S., Singh, B., & Sharma, S. (2015). Routing in wireless mesh networks: Three new nature inspired approaches. *Wireless Personal Communications*, 85(4), 3157–3179.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Shakti Kumar is Vice Chancellor Baddi University of Emerging Sciences and Technology, Baddi (HP), India. His research areas include Communication Networks, Soft Computing and Intelligent Systems. He has published 206 research articles in various Journals and Conferences and 9 book chapters. He has organized 7 international and 6 national conferences in the areas of intelligent systems and networks.



Amar Singh is a research scholar with Punjab Technical University, Kapurthla, Punjab, India. He did his M.Sc. (Computer Science) from Kurukshetra University, Kurukshetra, and M.Tech. (Computer Science and Engineering) from MMU, India. His research areas include Wireless Mesh Networks, Soft Computing and face recognition. He has published 19 articles in various journals and conferences.



Sukhbir Singh Walia has done his B.Tech. from Guru Nanak Dev University Amritsar, M.Tech and Ph.D. from Punjab Technical University (PTU), Kapurthla (Punjab) India. His research areas include Soft Computing, Education Management, Mobile Communication, and Wireless Sensor Networks. He has published around 25 research articles in various Journals and Conferences. Currently he is Director, International Resource Centre in Human Values and Professional Ethics at PTU.