Chapter 1

INTRODUCTION

Professional sports tournaments is one of the crucial economic activities which have been organized in various parts around the world. In present scenario of globalization, where majority of sport tournaments happen within a time span of one or two year's duration. A mega sport events like Indian Premier League (IPL), Olympic, world cup of football, hockey and cricket etc have a huge impact on viewers and thereby these events directly contribute to economies of a country which is organizing those events. So to organize these events we need a sports scheduling. Sports scheduling is the NP-hard problem. TTP is a one of the sport league scheduling problem where we need to organize the game for participating teams either at home or away and need to calculate the cost required for the schedule produced. To organize these sports events we need to have an efficient sport scheduling algorithm which can obtain better and effective schedule with lesser costs. We need to form an effective schedule, so that it attracts several viewers and thereby increase in revenue. A good schedule contributes to increase in revenue of a country and poor schedule leads to have a negative impact on the revenue. We used biogeography based optimization technique for schedule generation and hybridize it with enhanced simulated annealing to obtain the effective schedule. BBO is global optimization method which represents organism distribution in our biological system in terms of a mathematical model.

1.1 Motivation

Various researchers have applied different effective methods like constraint programming, scheduling theory, graph theory approaches and scheduling theory in order to solve the issues of sports scheduling problems. They hybridize one algorithm with other in order to create the optimal schedule which can represent as an effective solution for the TTP and other scheduling algorithms. Recent researches to the various scheduling problems like Italian football league scheduling, scheduling for New Zealand basket ball problem. Due to combinatorial nature of these problems several hybridizing approaches like integer programming and constant programming have been used in order to solve these problems and providing an effective optimal schedule in which overall cost is less for the match playing.

Good schedules are required in order to obtain huge amount of revenue from the match thereby considering the fact that intake cost includes money for organizing the match, distance travelled in terms of costs, money spent on players for their training during the match should be far less in comparison to the profit produced in terms of revenue. If that condition is satisfied then there is benefit in hosting the matches for the country because this leads to increase the revenue and directly contributes to the economic growth of the country. Sports scheduling need to map mathematics approaches to computer science and providing some operations that can be helpful to the economies.

The methods built for optimizing NP-complete and NP-hard problems which are not only combinatorial but related to nature of geography have motivated us towards these sports scheduling algorithms.

Nature inspired algorithms have been used to solve the most complex problems these days because these algorithms provides solution by considering some patterns and facts of nature and map it with some mathematics to produce desired solutions. To provide optimal solution for TTP and for its global convergence, this thesis attempts to explore swarm intelligence.

1.2 Related work

Several evolutionary techniques like integer programming, graph theory and nature inspired algorithms has been applied to TTP. In the research paper provided by Easton [1] , the TTP problem is usually described as one of the sports scheduling problem and correlated with Major League baseball. There are several benchmarks used to solve the problem which has been listed in [21]. Distance matrixes are included in these instances of [illegible] team where this distance matrix needed to be converted into a suitable input of our algorithm. If we are able to achieve maximum number of breaks then we can decrease the cost of schedule for constant instances.

Easton *et al* [12] hybridize integer programming and complex programming. They used the distributive approach and use 20 processors in parallel to run the code. By using these processors as recourses problem up to six instances are solved within a time of few minutes and higher team size takes longer time of few days. For NL 24 team size it takes about 20 days of work and computation.

Many researchers apply different combination of algorithms to provide efficient schedule for the TTP. Lagrange relaxation is hybridized with constraint programming by Benoist *et al*. [13] in order to develop a hierarchical model to improve the cost of TTP. The whole problem gets solved be constraint programming. Lagrange plays important role to solve the global constraints. Lagrange relaxation provides an effective optimization for solving these global constrains. One problem is divided into various sub problems and then Lagrange multipliers value gets modified for a particular team. TSP is a sub problem for TTP in which cost for the distance travelled should be minimum for a single team.

A hybrid approach consisting PSO-SA for TTP has been discovered by Alireza Tajbakhshl et al. [25]. PSO with SA achieved desirable results but there is one problem in it that its produced results have a delay of 2 hours with the best results known. ACO have been applied to TTP [11] but its convergence is uncertain. So to resolve this problem BBO have been applied to TTP [12] because BBO's convergence is certain. Their results are promising but they are unable to deal with dynamic constraints. Recently BBO is hybridized with ACO and PSO in the field of remote sensing as stated in [10]. BBO have evolved to next level as stated in [13] as extended BBO which considers some new factors responsible for evolution. The concepts of evolution and extinction of extended BBO motivate us to solve this NP hard problem with dynamic constraints.

1.3 Problem Statement

TTP is sports scheduling problem in which input is a distance matrix of n*n, where n is number of teams having some constraints and corresponding to that output is a double round robin tournament matrix which should be optimal in order to reduce the playing cost and their by increase in revenue.

Optimizing the schedule is economically related to the air fare cost for the teams who are moving to another location for match. More efficient schedule leads to profitable for the country who host the matches.

This thesis aims to provide a hybrid heuristic approach for solving TTP. The proposed system will use probabilistic measures to BBO in collaboration with enhanced simulated annealing to provide an efficient solution. The proposed approach hybridizes Modified BBO with efficient simulated annealing in order to solve the problem of local minima which is problem for several genetic algorithms. We can define our problem as:

"To develop a novel heuristic by hybridizing biogeography based optimization and enhanced simulated annealing based on extended species abundance models of biogeography for solving Travelling tournament problem."

1.4 Scope of the work

This project provides a solution for TTP problem by hybridizing modified BBO and enhanced simulated annealing. We hybridize it with enhanced simulated annealing in order to solve the problem of local minima. We did modification to immigration step of BBO which is capable of handling noise. That's why we collaborate our BBO with Kalman filter to work it under noisy environment. We compare our results to the previous algorithms which provide results for solving TTP and our algorithm converge to better optimal solution quickly.

Broadly scope of this work can be summarized as follows:

- To develop a fast hybrid heuristic for cost optimization of mTTP schedules.
- Adapt biogeography based optimization with its modification for development of fast constructive heuristic.
- To collaborate with BBO with kalman filter which has been done as a part of our modification.
- Modify immigration step of BBO, this generates efficient mirrored schedule for enhanced simulated annealing for further optimization.
- Use efficient simulated annealing to optimize schedules.
- To apply our approach to all extended species abundance models of biogeography.

- Compare the results with the best results available in literature.

1.5 Organization of the thesis

Remaining part of this thesis is organized in the following chapters:

Chapter 2: Problem Description and Computational Solutions

This Chapter includes all details of problem statement which specifies everything about TTP like its definition, constraints etc.

This chapter also explains the BBO algorithm which we are using for solving TTP.

Chapter 3 Swarm Intelligence for TTP

Then chapter includes literature survey in which we considered all previous researches on this problem and all the proposed methods developed to solve this problem till date. This chapter also contains the techniques which have been applied to TTP till today. These techniques use conventional approaches and hybrid with nature driven algorithms and with some mathematics in order to achieve the solution for TTP.

Chapter 4 Hybrid Heuristic for TTP

This chapter consist our hybrid approach. This chapter contains all the techniques and methodologies which we are using. All the details of probabilistic measures implementation on BBO and how its concept comes into existence of solving our problem have been explained in this chapter.

This chapter explains the system architecture of our proposed approach. There is block diagram of our system behavior which elaborates the functionality of our proposed algorithm and its working procedure. This chapter also includes enhanced simulated annealing approach for solving the TTP problem

We also give an algorithm which have been illustrated and explained in this chapter. It explains our approach in trying to solve Mirrored Travelling Tournament Problem and gives the idea how we are going to proceed to generate efficient mttp’s Schedules.

Chapter 5: Computational Results

This chapter illustrates the experimental setup used to obtain the results. All parameters specification and working principle of our code have been explained and comparison of our results with the previous versions has been presented.

Chapter 6: Conclusion and Future work

We conclude our work in this chapter. Scope of future result is discussed. Challenges faced by our problem and where improved can be done is highlighted.

Chapter 7: Publication from Thesis

This Chapter includes the details of the research paper, and its publication status.

Chapter 2

Problem Description and Conventional Solutions

2.1 Travelling Tournament Problem

TTP is the problem of obtaining schedule in which teams that are participating are listed and corresponding to that there are rounds which shows the location of team where they are going to play all of their matches. Our goal is to optimize this schedule by reducing the cost which is organizing cost at particular location. Firstly the question arises why there is need of modification in the current algorithm. We are dealing with TTP i.e. NP-hard problem, which is dynamic in nature so we have to modify our solution which can take care all dynamic constraints like problem of earthquake, floods etc. of nature.

2.1.1 TTP terminologies

Single round robin tournament: We have to form a schedule which is in the form of teams and rounds. Each team has to play a match with other team at least once in the tournament. Each team has to play one single match within a round with other team. The below figure represent the single team schedule in which the vertical column indexes are used for teams and horizontal indexes are used to represent rounds. For representation we have taken the team size as 6, therefore rounds are 5 in single schedule.

Table 1.1 Single round robin schedule

T/R	1	2	3	4	5
1	6	-2	2	3	-5
2	5	1	-1	-5	4
3	-4	5	4	-1	6
4	3	6	-3	-6	-2
5	-2	-3	6	2	1

6	-1	-4	-5	4	-3

So for n as a team size, the total number of matches played between them is n/2. So for a team with size n total supported round should be n-1 in which each team play a match with the other team

Double Round Robin Tournament:_ Double round robin schedule is created by taking the mirror images of the single round robin schedule providing that the teams with matches as homes in first half gets converted into away in second half and the teams with matches as away in second half gets converted into homes in second half. The above method described has been shown below:

Table 1.2 Double round robin schedule

	First Half					**Second Half**				
T\R	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***
1	6	-2	4	**-3**	-5	-6	-2	-4	3	5
2	5	1	-3	**6**	4	-5	-1	3	--6	-4
3	-4	5	2	**1**	6	4	-5	-2	-1	-6
4	3	6	-1	**5**	-2	-3	-6	1	-5	2
5	-2	-3	6	**-4**	1	2	3	-6	4	-1
6	-1	-4	-5	**-2**	-3	1	4	5	2	3

2.1.2 TTP constraints which have been used are:

We use two constraints for the TTP problem:

1. **Maximum**: No team is allowed to play consecutively 3 games in home or away. Let ▯▯ be the number of games by team I then three cases have to be considered.

Case 1: If (▯▯▯ ▯ ▯▯▯)

1.) If □□ played its last game in its home location then our assignment strategy should assign next game in □□ home location.

2.) Otherwise schedule game in □□ home location.

Case 2: If (□□□ □ □□□)

1) If □□ played its last game in its home location then our assignment strategy should assign next game in □□ home location.

2) Otherwise schedule game in □□ home location.

Case 3: If (□□□ □ □□□)

1) If □□ played its last game at away location and □□ played its last game at home, then schedule the next match between them at □□ home location.

2) If □□ played its last game at home location and □□ played its last game at away, then schedule the next match between them at □□ home location.

3) Otherwise, use random function to assign either at □□ or □□

2. **Without repetition**: If □□ and □□ played a match in □□ location then second match between □□ and □□ is not going to be played at □□ home location.

2.2 State of art of TTP

Approach of simulated annealing has been proposed firstly by Anagnostopoulos *et al.* [17], which did the revolutionary thing for TTP. In his approach he used various complex moves in order to obtain neighborhoods. Both infeasible and feasible schedule have been explore through the mechanism of searching effectively. The problem of local optima gets solved by concepts like strategic oscillation and reheats. Several variations have been applied to the complex moves in order to obtain neighborhood for TTP.

Iterative local search have been proposed by Costa *et al*. [18] which comes out to be a good technique. In the ILS approach there are two types of permutation and two moves is used. Most res use the polygon method to generate initial schedule but they didn't use it. They apply canonical-1 factorization method for generating the initial schedules and replace the polygon method. This method is successfully applied to various sub problems of TTP to generate initial solutions. With the computed result it has been proven that ILS is better than integer programming and also has a good convergence rate toward problem.

Later ILS is hybridized with GRASP (Greedy randomized adaptive search procedure) by Ribeiro *et al* [8] and proposed a new optimal heuristic for TTP. They used roulette wheel selection method in order to generate initial schedules. An abstract schedule gets created on the basis of from initial random permutation. Now this abstract schedule gets converted into real schedule by using mapping functions. Schedule gets improved by a margin. After that stadiums have been allocated to teams and a matrix is obtained for it. These schedules are better than random schedule because effective mapping function have been used for them. They used four neighbor structures on which the hybrid technique of ILS and GRASP works. To reduce the complexity further ejection chain mechanism has been used with it. These results are revolutionary and took 1 hour for the computation of the larger instances. They set a new record which beats the all previous records.

Urrutia *et al.* [19] have invented new methodologies which are better and beat the results provided by Easton *et al.* [1]. Each team needs to travel a particular distance which is considered as the difference between minimum numbers of road trips consecutively required for the completion of tournament for instances with optimal solutions.

Later Tabu search is used to provide optimal schedule for the TTP by Gaspero et al. [20] which is considered as good . In his work approximate solutions can be generated with the help of tabu search. He uses several different and complex structures for providing the initial schedule for the problem instead of permutations. He compared the achieved results with the previous results and found quite improvement in it

Various combinations of different techniques have been integrated with SA due to its global convergence nature. One of the methods proposed by Lim et al. [21] is presented in literature. In work proposed above, the combination of SA and hill climbing was introduced. Two parts of search space was used one is team assignment space and other is timetable space. The hill climbing algorithm controls team assignment phase while simulated annealing manage time table space. Time tables generated by simulated annealing are better. This time table so produced is transferred to the hill climbing module, where further modifications and minimizations are applied to it and a better team assignment strategy is applied to it. The return value of this module is optimal team assigned schedule to the simulated annealing component. This process keep on going until an effective schedule is generated or loop gets terminated. The complete reason of this hybridization is: try to make optimal schedule and team assignments only when the linked timetables to them have a probability of providing better solution exists. This saves a lot of computation time.

Hentenryck *et al.* [22] proposed an effective method for solving TTP in less amount of time. Most the better results known today have been introduced by Hentenryck. He improved the previous described approach taken by Anagnostopoulos *et al.* [17]. He give a modification to the objective function and considered it as soft constraint and there is also a penalty for missing this constraint. This modification improves the way of exploring large region into an efficient schedule. The results achieved by this technique are more promising.

This section consist description of various techniques that have been applied to TTP in order to provide optimal solution. We also elaborate their strengths and weakness of their approach used for problem solving. From the year 2001 onwards, the work has started for providing optimal solution for TTP. During these 13 years lots of researches has been done on order to solve this NP hard problem. Many researchers have successfully solve the problem for smaller instances but for larger instances like NL team size 16 etc is yet to be solve efficiently. These techniques provide solution for TTP for smaller instances very quickly but for large value of team size, they need time to provide solution and solution is not optimal.

We categorize the approaches used to solve TTP in two main categories:

1. Conventional approaches
2. Nature inspired algorithms

Here we discuss conventional metaheuristics and its working in order to solve the TTP problem. Nature inspired algorithms have been discussed in next chapter.

2.2.1 Conventional metaheuristics

These techniques have concepts of programmable algorithms which have been widely used for problem solving and either they have been derived from mathematics directly (like integer programming, constraint programming, tabu search) or inspired by simulated annealing which is better than genetic algorithm and better convergence rate. These techniques have been applied to variety of problems that have high complexity and these methods show effective results. Below we mentioned some of them which provide their way of solving the TTP.

2.2.1.1 Tabu Search

Tabu search comes into existence in the research for TTP in 2006 by Di Gaspero. He found the good results at that point of time. In his work, a group of tabu search solvers proposed the approximate solution for TTP. At that time researchers use neighborhood structures with their complex combinations are collaborate together. These different neighborhoods are analyzed by various technologies and compared on the basis of simulation results experimentally. The researchers evaluated their results on several sets of benchmarks and their results were compared with previously known results as discussed in the literature.

a) Shortcomings of tabu search

Di Gaspero recorded that tabu search is comparatively much slower than simulated annealing in order to perform every single iteration due to the problem of exhaustive exploration found in the complex combination of neighborhood structures. In order to provide the effective results tabu search needed to be correctly implemented and with in proper experimental setup. In their case, lot of things needed to recalculated and should be changed, especially the cost regarding the computation between two neighborhood structures.

2.2.1.2 Combined integer and constraint programming

This was the hybridized solution after integrating of various methodologies like graph theory, genetic mathematics and evolutionary computation proposed for TTP by its creators Easton et al. [12]. Their proposed solution for the TTP is a branch-and-price (column generation) algorithm in which they represent individual team tours as the columns. In branch and price algorithm, the root node of the branch contains the linear programming (LP) relaxation and some small subset of columns arte the parts of bound tree. A sub problem was introduced which represents the LP objective, called as pricing problem, which gets provides solution for the issue whether there exists any additional columns to be integrated in base solution. LP is re-optimized if the returning variable of pricing problem has value greater than equal to one, that means one or more columns exists. The algorithm branches if value returned by returning variable is zero and the LP solution are fractional.

Branch-and-price is an aggregation and generalization of branch-and-bound with LP relaxations. From the two approaches constraint programming and integer programming approach, they prefer constraint programming to solve the pricing problem.

a) Shortcoming of Combined Constrained and integer programming approach

When instances n = 4 have taken, problem gets nearly solved. With instances n = 6 are more difficult and challenging. The proposed models without parallel programming solve these instances in a reasonable and affordable amount of time. With the help of parallel programming if 20 processors have taken to solve instances with n = 6,the calculated computation time is very less as compared to single processor and it is directly proportional to order of minutes □ □□ □□□ At last they found that parallel programming is necessary for solving instances with n = 8 teams. For 20 processors, it takes approximately 4 days to solve these problems.

Its get proved that for smaller instances parallel programming can be avoided, but for larger instances without parallel programming optimal results can't be achieved. So we need to use parallel programming for larger instances to provide an efficient solution.

2.2.1.3 Simulated Annealing

This technique is Proposed by Anagnostopoulos [17], further which gets analyzed and improved in [22], simulated annealing quite successful to achieve optimal results for solving TTP quickly because convergence rate of simulated annealing is very fast than other algorithms. A hybrid algorithm based on the metaheuristic (simulated annealing) proposed by Anagnostopoulos for exploring both feasible and infeasible schedules in TTP. The hybrid approaches combine the advantages and principles of other metaheuristics as: it uses complex moves having concept of swapping the teams and rounds and output a large neighborhood because of inclusion of advanced complex techniques such as reheats for escaping from problem like local minima at very low temperatures and strategic oscillation for balancing the exploration of the infeasible and feasible regions. It compares the well known solutions and their problems as well results for smaller instances and for larger instances. The worst calculated solution value produced by simulated annealing algorithm over 100 runs is far less than or equal to the best known solutions of other algorithms for TTP because of its robust nature.

a) Shortcomings of Simulated Annealing

Although simulated annealing is better technique but there is a problem that is its computation time which can degrade its performance. For real life problem the computation time requirement for this technique is a great issue which makes SA infeasible to such problems. It has a delay of 54 hours for a team size 16. So it is needed to optimize. That's why we upgrade this technique.

2.2.1.4 Iterated Local Search

ILS have been researched and developed by Urrutia *et al.* [10]. In their work, they proposed hybrid heuristic which combine the approaches and principles from ILS and GRASP metaheuristics for getting solution for mirrored TTP. Initial step/schedule has been constructed by a three step algorithm which has been discussed step wise hare. In the first step, for constructing a timetable, they used the method called as canonical 1-factorization with placeholders. In the second step to assign teams to placeholders greedy heuristic is used. In the last step of the constructive heuristic, the games venue are set round by round and to repair possible infeasibilities local search is used. Four simple neighborhoods have been used in the hybrid heuristic for doing local search and for perturbations; one ejection chain neighborhood is used. The results by doing this hybridization are far better to some instances for TTP at that time. It has been proven that the constructive algorithm is very quick, fast and produces good initial solutions/schedules that can lead to a better solution and quality of results have been improved.

a) Shortcomings of Iterated Local Search

This technique is good but doesn't able to beat the best results which have been found in literature. This technique produces a gap of 17% which is not allowed. This gap is more and should be needed to reduce. Our objective is to reduce the cost of the effective. Schedule as much as we can and this gap should get reduced in order to complete the objective of finding minimal cost for the TTP.

Chapter 3

Swarm Intelligence for TTP

TTP is a complex problem in field of evolutionary computing and integer programming. This problem has a relationship with TSP problem in which task is to minimize the total distance travelled. Many trials and attempts of different types of hybridization for TTP have been proposed. In this section we talk about all swarm intelligence techniques developed for providing an optimal solution for TTP.

3.1 Nature inspired algorithms:

In this section which has been provided below, we discuss few algorithms which are in categories of genetic algorithms and talked about the results provided by them in order to solve every issue of TTP. These methods are nature driven and therefore directly dependent on the organization and way of interaction and their behavior of animals (Ant colony, bee colony), which help in evolution and existence (genetics , biogeography) in order to develop social behaviors (bird flocking, fish schooling) etc. The advantage of these algorithms is that for a given problem they achieve the objective to find a near optimal solution. These algorithms show quite good results but a few text and literature have been written for their existence as compared to conventional metaheuristics.

3.1.1 Genetic Algorithm

Evolutionary computation has been adopted as the computation which can provide a proper solution for TTP which was first used by Biajoli *et al.* [24]. The methodology used a hybridized approach of incorporating genetic algorithm with simulated annealing. The idea behind this hybridization is to provide the initial solution by genetic algorithm by cross over concept and local search implementation for feasible schedule by simulated annealing for obtaining better schedule.

In the work proposed in which genetic algorithms implemented collaborating with simulated was annealing in order to solve local minima problem. The relationship between the local search and evolution, genetics and learning has been established by this hybridization. Learning is correlated with local search for the near optimal solution and for each individual some modifications has been done in order to learn the behavior from the ancestors. Genetic algorithm made local search more effective and increase the probability of convergence to feasible schedule. For the application of the GA, a compact complex representation was proposed for the chromosomes (individuals). The chromosomes use the algorithm of code expansion in order to populate.

a) Shortcomings of Genetic Algorithm

This approach provides a gap of 12% from the best results known. So it is also unrealible for problem like TTP which is a real life problem

3.1.2 Particle Swarm Optimization

A hybrid approach consisting PSO-SA for TTP has been discovered by Alireza Tajbakhshl et al. [25]. In the hybridization approach two techniques were used first one is particle swarm optimization and the other is simulated annealing. Particle swarm optimization is to generate initial schedule and to apply local search on that schedule they used simulated annealing. This PSO is based on the 0-1 logic for generating the optimal schedules for TTP. The second phase of hybrid algorithm includes SA which is applied to make feasible schedule and to solve local minima problem.

SA improves the schedule obtained by first phase. For team size 4, 6, 8 it is able to generate proper schedules with low overall costs.

a) Shortcomings of PSO

A mathematical model proposed by Tajbakshi provides some way to think a solution for TTP in efficient way. Their results are revolutionary but they still takes 7200 sec (2 hours) to compete to the best solution. We can summarize from this chapter that there are various techniques which have been applied in hybridization with each other for providing the best results. This hybridization may not give the best but it’s still better for smaller instance and beat the records of the conventional heuristics. Now let us explore various nature inspired algorithms and choose the best fit for our problem.

3.1.3 Ant Colony Optimization

In ant colony optimization searching agents are called artificial ants. These ants are used for searching purpose. Collectively these software agents search optimal solutions for the given problem and this solution is optimal. ACO uses the algorithmic approach of finding the best path for a weighted graph like data structure algorithms (warshall's algorithm). The solution made by these artificial ants is based on incremental approach in which by moving on graph they build solutions incrementally. Pheromone model is the basis of solution construction process and this process is stochastic and biased. The value modified at run time by ants for the set of parameters which are associated with graph components. A hyper heuristic which is ant based for solving TTP proposed by Chen *et al.* [26]. In the proposed model every low level heuristic represents set of vertices. For carrying initial solutions all these hyper heuristic agents locate themselves uniformly around the vertices of a network. The working of each ant is to traverse along the edges present in the network and needed to reach to another vertex after reaching to another node the low level heuristic approach is applied by the software agent called ant on that node. In the graph there is no restriction on number of times an ant visit on a particular node. Ant can repeatedly apply the same logic of low level heuristic on the particular node which it had visited in past. The two concepts pheromone trails and visibility is used in combination to each other. The computation time for a particular vertex by the heuristic is termed as visibility. Good quality heuristic is preferable for providing better solutions.

a) Shortcomings of ACO

1. It produces difficult theoretical result.

2. For each iteration probability distribution varies

3. Sequences need to be dependent, but they are independent.

4. Theoretical proof is difficult to implement

5. Its convergence rate is uncertain and takes a long time.

By Application of ant colony optimization the results obtained are not optimal although logic of ACO as a hyper heuristic. The results were even worse for larger instances. ACO advantage is that they are able to produce results for each instance whether it is good or poor.

This chapter consists of various techniques comes under category of conventional heuristic and in nature inspired algorithms. After exploration of these algorithms we come to a conclusion that BBO is still worth full if apply in a proper way for solving the TTP problem.

3.1.4 Biogeographically Based Optimization

BBO is an evolutionary algorithm whose working principle is based upon migration mechanisms of species from one habitat to other depending upon the fitness of the habitat which are favorable to them. It is a mathematical model for implementing organisms or species distribution in biological systems. This concept was first introduced by Dan Simon. The habitat which have high HSI (high suitability index) have high value of species count. Therefore habitat which has high value of HSI have high emigrating rate i.e. it is ready to send its SIV to other habitat, while the habitat having low value of HSI have low value of species count and their immigrating rate is high, i.e. it is ready to accept species towards itself. HSI of a habitat can be affected on the basis of SIV (suitability index variables) which are independent variables. We are applying BBO after integrating it with our approach and hybridizing it with enhanced simulated annealing to produce an optimal solution of TTP.

3.1.4.1 Principle approach

BBO uses the term called as habitat which is the home location of the individual. Species can migrate from one geographical area to another on the basis of high suitability index (HSI).HSI can be influenced by various factors such as rainfall, earthquake. Different variables that affect the HIS of a particular region are known as Suitability Index vector (SIV). HSI are dependent Variable because it depends on various factors while SIV is considered as Independent Variable.

Habitats which have high value of HSI support more number of species as compared to habitats which have low value of HSI. If population of a habitat keep on increasing then species migrate from current habitat to the habitat which is more promising to them. Just few species are allowed to move so emigrating species has its contribution to both home and emigrating habitats.

Therefore High value of HSI habitats have high emigration rate. As population of high HSI habitats is high so immigrating species are not allowed to move here. Therefore if a habitat has low HIS have low emigration rate and high value of immigration rate. The above relationship can be shown by the figure given below.

3.1.4.2 Working Procedure of BBO

The below diagram illustrates the basic mechanism and relation between immigration curve and emigration curve. Here I is maximum Immigration rate, E is maximum emigration rate, S_0 is equilibrium number of species, μ is emigration rate and λ is Immigration rate.

Immigration curve :

- *I* is the Maximum possible immigration rate for a particular habitat.
- When number of species in a habitat is zero then only maximum immigrating rate I can be possible.
- Immigration rate is inversely proportional to number of species.
- Immigration rate reduces to zero, when count on the number of species reaches to its maximum value for a particular habitat.

Emigrating Curve :

- Emigration rate reduces to zero when no species exist in habitat.
- Emigration rate is directly proportional to the value of HIS of a habitat.
- When the population reaches to its maximum capacity then value of HIS gets maximum and therefore maximum emigration rate E has been achieved.

The state of Equilibrium :

- S_0 is the equilibrium number of species. Immigration and emigration rate becomes equal when value of count reaches to S_0 .

- After a certain amount of time every habitat gradually reaches to the state of equilibrium.

- Deviation of curve from point of equilibrium starts once it reaches there because of following two reasons.

Positive excursion: It could happen due to some catastrophic event occurring in neighboring habitat.

Negative excursion: It could happen due to introduction of some unexpected predator.

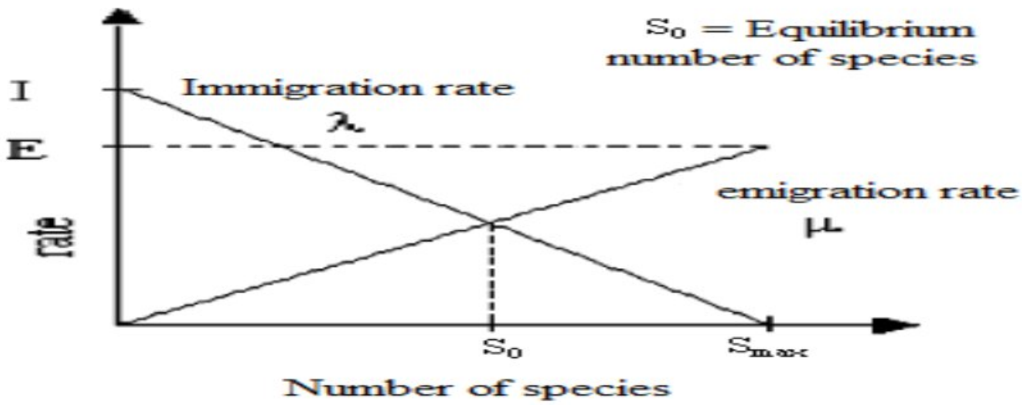


Fig.1 BBO working principle

3.1.4.3 Operations in BBO

There are two type of operation that can exists, these are

- Migration
- Mutation

a) Migration

It is the process of migration of one SIV to the other supporting habitat. Let us suppose there are two candidate solutions S1 and S2.S1 is bad while S2 is good one. Number of species in S1 habitat due to its bad feature is lesser than the number of species in S2 habitat. Fitness or HSI of a particular solution is converted to the number of species it contains. The immigration rate of poor one (S_1) has a high value than better one (S_2) *i*.e. $\lambda_1 > \lambda_2$ and emigration rate of S1 is lower than better one *i*.e. $\mu_1 < \mu_2$.

If we want to make a comparison between relative migration of above candidate solution then can be illustrated from the below figure:

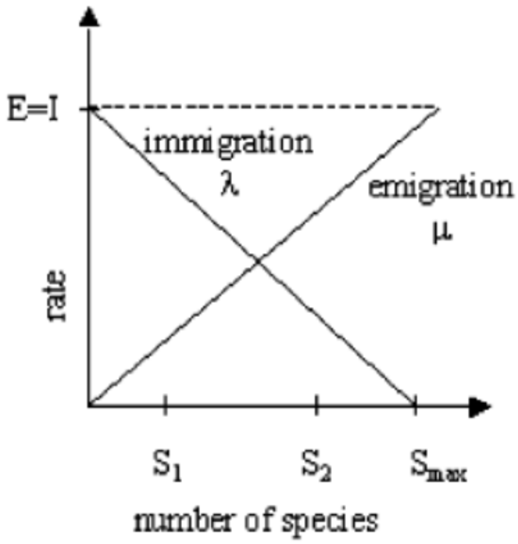


Fig 2 Migration mechanism

If we have fitness value for a solution, then we can calculate its number of species relative to other solutions. And if we have count of number of species, then we can calculate λ and μ which are basic parameters of BBO.

Migration algorithm:

1. Select habitat □ □□ with the Probability λ □
2. If □ □□ is selected
3. For j=1 to n
4. Select □ □□ with the probability □□.
5. If □ □□ is selected
6. Use rand (0, 1) to select SIV □ from the □ □□.
7. Replace random SIV with □
8. End of if
9. End
10. End of if

b) Mutation

Habitats can change their fitness value by random cataclysmic events. These events are very rare but their existence cannot be ignored and must be considered as a interruption in BBO model. These events lead to change the state of a habitat to shift from its equilibrium state. Mutation is the process of reflecting cataclysmic events in BBO. As these events are very rare, therefore the Probability of mutation is assigned to a very small value. After considering the comparison of fitness of candidate solution we analyze that probability of occurrence of very low number of species and very high number of species is low. Natural habitats have steadily and gradually evolution rate. If we considered any set of habitats, most of them show similar behavior of having number of species with in some defined range of equilibrium. This state can be achieved after a long time since the habitat came in to existence. Thus at any given point in time, each population member has an associated probability which tells that it was expected as a solution. If a habitat Hi with probability Ps has currently very low number of species or very high number of species then it is surprising that it exists. The whole reason of doing mutation is to increase diversity in population.

We assume that it is due to some cataclysmic event and thus we like to mutate it. So the probability of mutation is high for very good and very poor solutions and low for mediocre solutions. Mutation probability is defined by the equation

$$m(s) = m_{max}\left(\frac{1 - P_H}{P_{max}}\right)$$

Where

$m(s)$:is mutation probability;

m_{max} is the user defined parameter (maximum value of it is. $0 \leq m_{max} \leq 1$.

P_{max}: is the apriori probability of occurrence of the best habitat.

P_H: is the apriori probability of occurrence of habitat H

BBO is evolutionary algorithm which has hybridized with Simulated Annealing [12] for solving TTP. But that model will work only if we don't consider noise. That means the nature becomes static. In real applications dynamic constraints exists like earthquake, floods etc which can affect the fitness of the habitat. To deal with dynamic constraints some features are required to integrate with existing structure. A new approach towards BBO algorithm is proposed in [13], which redefines the concept of BBO and evolves a new strategy based on extended species abundance models of BBO. In [13] concept of growth rate and decline rate is introduced. The next section explains the extended models of BBO.

3.1.4.4 Extended Models of BBO

In previous section BBO has been explained, but here we are explaining extended version of BBO [13] which we are using in order to solve Travelling Tournament Problem.

Extended BBO proposed six different models. The algorithm introduces an additional dependency factor which signifies the interdependence of migrating species on each other such as the predator-prey relationships to be considered for the determination of immigration and emigration rates and hence modifies the Simon's model of species abundance in a single habitat. The models are:

Initially BBO defined by Dan Simon is taken as with main parameters as Immigration rate and emigration rate, where

□ □ $\frac{□}{□}$ □□ ∗ □□□ (22)

□ =□∗ □□ $\frac{□}{□}$□□ (23)

Which get modified on the basis of six models which is stated as :

a) Linear models

Model 1: (Constant immigration and linear emigration model):

Here we used the constant immigration rate and emigration rate has to be taken as linear. There are equations for Growth rate and Decline rate as follows:

Growth Rate □ = 1/2+c/k; (24)

Decline Rate □ = Ek□/n + 2√□□; (25)

Model2.(Linear immigration and constant emigration model):

Here we used linear immigration rate and constant emigration rate.There are equations for Growth rate and Decline rate as follows:

Growth Rate □ =I(1-k/n)+c/k; (26)

Decline Rate □ =E□/2+2√□□; (27)

Model3.(Linear immigration and linear emigration model):

Here we used linear immigration rate and constant emigration rate. There are equations for Growth rate and Decline rate as follows:

Growth Rate □ =I(1-k/n)+c/k; (28)

Decline Rate □ =E□k/2+2√□□; (29)

b) Non linear Models

Model4.Trapezoidal model:

If (k<=i)

Growth Rate □ = I+c/k; (30)

Decline Rate □ = Ek□ + 2√□□; (31)

Else if(k<=n)

Growth Rate □ =2I(1-k/n)+c/k; (32)

Decline Rate □ =E□+2√□□; (33)

Where i' is the smallest integer that is greater than or

Equal to □□□□/□ □ □□h□□ □□ □ □ □□□□ □□□□/□ □

Model5 (Quadratic model):

Growth Rate □ =I□□/□ □□□+c/k; (34)

Decline Rate □ =E□□/□ □□+2√□□; (35)

Model6 (Sinusoidal model):

The migration rate □□ and λ□ are sinusoidal functions of the number of species k, resulting in a bell-like shape.

Growth rate = □ = □/□(cos(□□/□)+1)+c/k; (36)

Decline rate □ =E□ (-cos(□□/□)+1)+ 2√□□; (37)

3.1.4.5 Effect of noise on BBO algorithm

Let we consider two habitats [illegible] and [illegible]. These habitats have their fitness's as [illegible] as immigrating and [illegible] as emigrating habitat. Let noise involved in two habitats is [illegible] and [illegible]. Due to affect of noise the measured fitness is [illegible] instead of [illegible]. If we consider [illegible] has more fitness than [illegible] and let n1 has huge value than n2 and both high value than [illegible] and [illegible]. Therefore the overall fitness becomes:

[illegible] (1)

[illegible] (2)

[illegible] [illegible]

Therefore HB1 accepts the SIV from HB2, therefore condition of BBO gets satisfied as immigrating habitat fitness is less than emigrating habitat. But population of HB1 is already high due to its high HSI. The BBO migration procedure will corrupt. Its measured fitness is more if don't consider noise, so this immigration should not be done. That's why we need to modify it.

In order to calculate the uncertainties, we use the concept of noisy BBO [5].

[illegible] (4)

[illegible] (5)

[illegible] (6)

Where U is the uncertainty of the state estimate, m is the estimated fitness, z is the measured fitness, [illegible] is the variance of the process noise, and [illegible] is the variance of the observation noise. The uncertainty [illegible] and the estimated fitness [illegible] are the values from the previous iteration step before the most recent fitness measurement is updated. The process noise is assumed to be zero, therefore the uncertainty U is only related [illegible] and [illegible].

[illegible] (7)

[illegible] (8)

Because [illegible] 0, now [illegible] +1 > 1. Therefore [illegible] +1< [illegible] .With each step in the above modified algorithm, the uncertainty U will be reduced according to [illegible] and [illegible]. Small value of uncertainty leads to high accuracy of estimated fitness.

If limit tends to infinity, our purposed approach gives an estimate value of the fitness which is equal to the real value.

The next chapter of thesis represents our proposed solution for the TTP. It will cover our architecture and algorithm.

Chapter 4

Hybrid Heuristic for TTP

In this chapter we explain the working of modules of the approach. The next chapter includes the details of proposed system architecture and our algorithm.

When noise is involved in a system, all measured values in this system become the combination of real values and noise. Therefore the measured values are not accurate, and they cannot be used to repeat the real internal status of the system. The following example aims to demonstrate how noise changes the immigration and emigration rates in the BBO algorithm and damages the function of immigration in BBO.

The theory of the Kalman filter was invented by R. Kalman in 1960 [26]. It is a recursive filter which can estimate states in a noisy environment [27]. In the past 50 years, the contribution of the Kalman filter in noisy environments has been significant, and it has become the theoretical foundation of many famous applications, for example, navigation systems .One of the most important contributions of the Kalman filter is that it can make an estimation of the true state value in a noisy environment. In the BBO problem, each fitness is the sum of the true fitness and a random noise. Therefore the measured fitness are not equal to the true fitness. According to The detrimental effect of noise is that it changes the true emigration and immigration rates of each habitat. The Kalman filter provides a better estimate of the true fitness of the habitats compared to the measured ones. In this application of the Kalman filter to BBO, noise was added only to the fitness measurement, and this is called the observation noise. No noise was added to the system process.

We modified immigration step of BBO and hybridize it with enhanced simulated annealing based on extended abundance models of biogeography which can be explained in later section..

Our proposed approach is able to deal with dynamic constraint which create problem for BBO working under noisy conditions.

Our problem is to deal with dynamic problems, to refine the schedule, minimized the cost of the schedule for TTP and to solve the problem of local optima. For solving these issues we proposed an approach to add some features to the immigration step of BBO and then apply enhanced simulated annealing to the seeds produced by BBO and apply this approach to all extended species abundance models [11] and compare our results to the previous obtained solutions. We made a hybridized heuristic of modified BBO with enhanced simulated annealing which is used to solve issues regarding the sports scheduling algorithms and providing an efficient and optimal solution than other algorithms like ACO, PSP etc.

The purpose of modification of BBO is to deal with all dynamic constraints like earthquake, flood, tornado etc which can degrade the obtained results. We include probabilistic calculations in the immigration step of BBO which is able to increase the performance of our approach and able to handle the dynamic constraints effectively and converges to feasible solution very quickly. Its hybridization with enhanced simulated annealing provides a solution to solve the problem of local minima which is the basic problem of algorithms like ACO and PSO.

4.1 System Architecture

The architecture for our approach is represented in fig 3.

- Our input is a distance matrix of N*N, where N is team size. This team matrix contains the cost of travelling from one location to other as teams have to move away to play the matches at challengers venue.
- Our algorithm is evolutionary algorithm that works on migration mechanisms of species, so we need to map given input to this problem.
- For mapping we need to calculate the HSI for each habitat. We map the total cost which is given to us in terms of matrix to HSI by using mapping function stated in Table 4.
- We need to calculate species count because it is the parameter for calculating the growth rate and decline rate.
- We go for the modification phases for the immigration step.
- We apply enhanced simulated annealing for solving the problem of local minima and to converge our solution quickly. Enhanced simulated annealing has various stages. First it produces the neighborhood by using swapping moves. Then we apply standard simulated annealing algorithm in order to obtain feasible schedule. After those concepts of reheat and strategic oscillation is used in order to improve results.

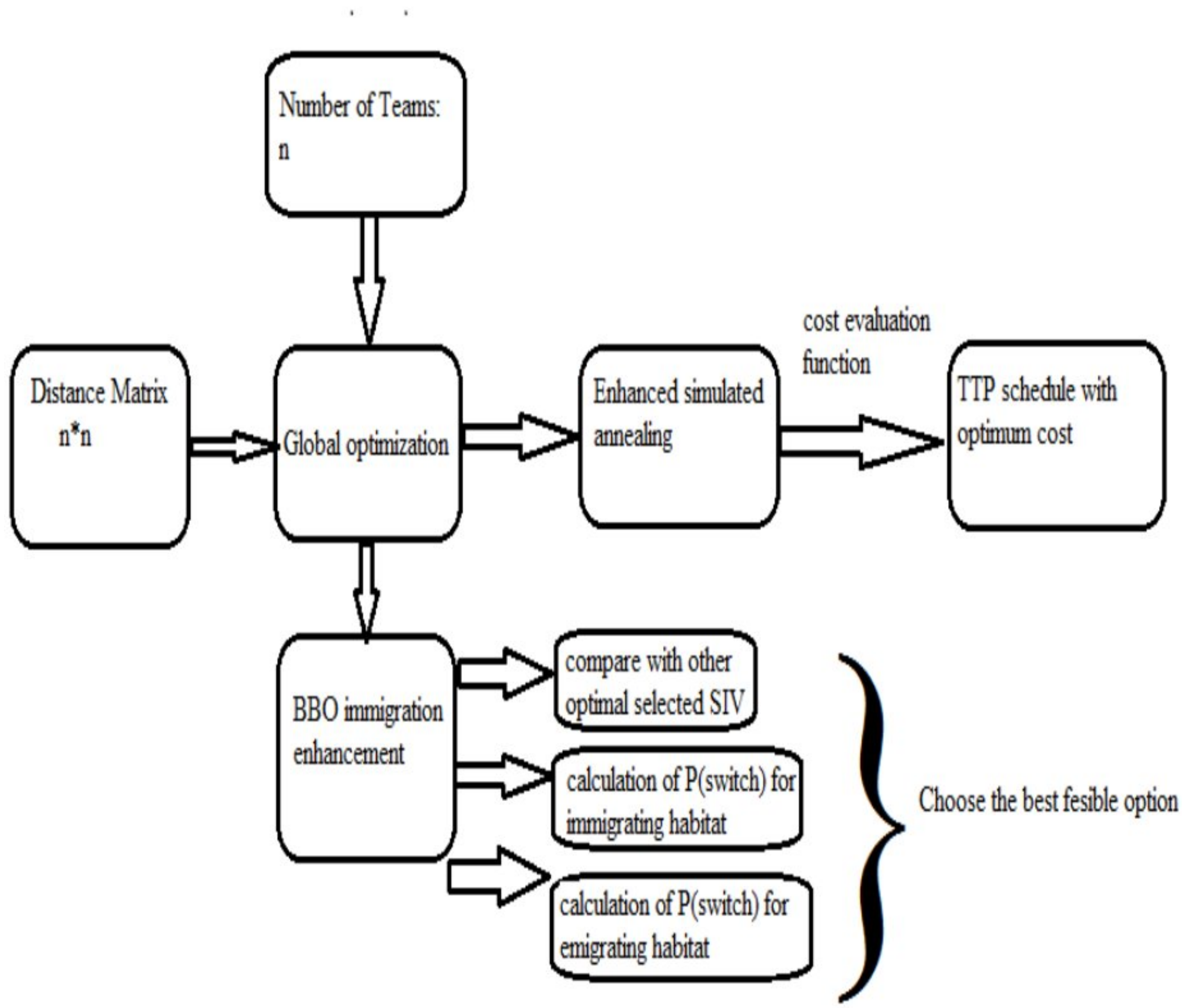


Fig 3: Proposed architecture

[illegible]

[illegible] (10)

[illegible] Species count, [illegible] Total number of species, [illegible] is dependency factor, I = Maximum immigration rate, E = Maximum emigration rate.

After getting the growth rate we go for modification phases to select the SIV which is going to be replaced. We use the decline rate in order to select the habitat which is going to send its SIV. After the implementation of our algorithm to the problem, we hybridize it with efficient simulated annealing to obtain the efficient schedule and to solve the problem of local minima.We implemented both linear models and non linear models of extended species abundance models to check the correctness of our algorithm. We compare our approach with the existing ones by testing it on matlab platform. We did simulation to calculate performance of our approach over 100 generations.

4.1.1 Efficient Migration Mechanism of BBO

We add some feature to the migration step of BBO by using some probabilistic measures. Our added features composed of three phases for optimizing the results.

1) Compare with the other optimal selected SIV

In this phase we consider two habitats h[illegible] as immigrating habitat and h[illegible] act as emigrating habitat. We consider two instances of h[illegible] as h[illegible] and h[illegible]. Firstly h[illegible] is going to accept optimal SIV from h[illegible] and then h[illegible] accept another best suitable SIV from h[illegible] and after that their performance gets measured based on the probabilistic measures as:

[illegible]

[illegible] (11)

[illegible] It tells the probability of a habitat h[illegible] with fitness [illegible] after accepting selected SIV greater than fitness [illegible]

2) Calculation of [illegible] for Immigrating Habitat

In this phase we calculate p(switch) from equation (15) which can be calculated as a integration to the function of [illegible] from [16] and [17]for immigrating habitat. These are two probability density functions whose value is computed by equation [15]. If the new estimated fitness of immigrating habitat after re-evaluation is still worse than emigrating habitat, migration is going to occur, else migration don't exists.

[illegible] (12)

3) Calculation of [illegible] for emigrating habitat

In this phase we calculate p(switch) for the emigrating habitat

[illegible] (13)

From the above phases we choose the best option for the immigration step.

Secondly we map this approach to all the variants of extended species abundance models of BBO and implement it on TTP problem. We modified the immigration step and apply this modification approach to all the linear and non linear models of BBO to check whether we are able to achieve the optimal results or not. We test our algorithm to obtain various results which provide optimal solution for TTP problem

4.1.2 Equation Used

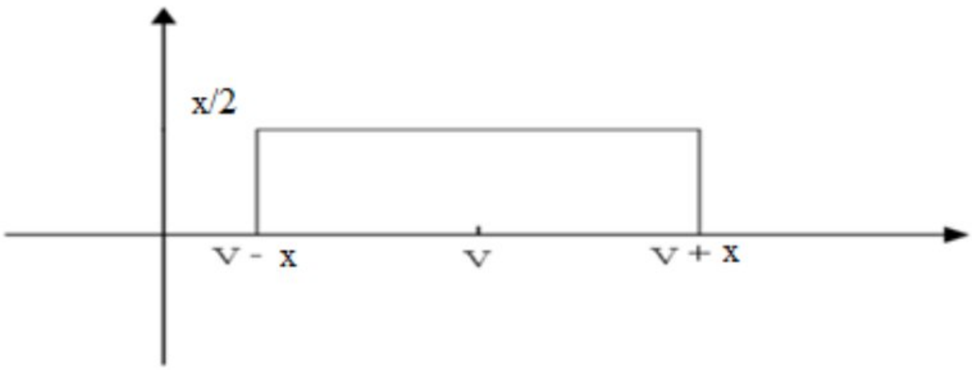


Fig 4 Probability Distribution function of fitness of habitat

[illegible] (14)

[illegible] (15)

[illegible]

[illegible] (16)

[illegible] (17)

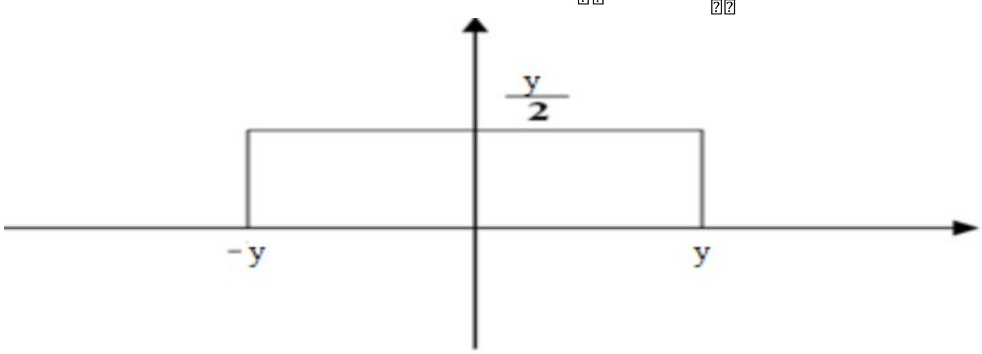


Figure 5: The PDF of noise involved in the fitness.

[illegible] (18)

The above equation is generalization of Baye's rule.

[illegible] Probability of a habitat with fitness [illegible] after accepting selected SIV greater than fitness [illegible] given that [illegible].

[illegible] where P(switch) is given by :

[illegible] (19)

[illegible]; Where y,x,z are noise and fitness range parameters

The PDF of p is as follows

[illegible] (20)

The PDF of q is as follows.

[illegible] (21)

This is the modification approach which is required to enhance the performance of BBO and to handle the dynamic constraints effectively. Next we talk about enhanced simulated annealing which is required to solve the problem of local optima.

4.1.3 Enhanced Simulated Annealing

This research project extends the simulated annealing as enhanced simulated annealing. The functional architecture of ESA has represented in fig 6 to achieve desirable goals listed below.

a) ESA used the tow categories of constraints one is hard constraints and the other one is soft constraints. Hard constraints are those which are always satisfied by the configurations and soft constraints may or may not satisfy by the configurations. The double round robin schedule is basically infeasible which may not satisfy the three constraints like maximum etc which have discussed in earlier chapter; therefore it is required for ESA to converge to a feasible solution.

b) ESA used a large neighborhood having time complexity as $O(n^3)$, where n is the team size. Because of swap moves it use 4(n-2) configurations.

c) ESA uses the objective function which is responsible for the minimizing of spent time with in feasible and infeasible region.

d) ESA use the reheats concepts as its one of the variants which is used to solve the problem of local minima.

4.1.3.1 Obtaining Neighborhood

Second half of the schedule is just the mirrored image of first half in provided a constraint i.e. the team which plays home in first half, plays away in second half and vice versa. ESA obtains the effective neighborhood which is a double round robin schedule which gets generated with the help of moves mentioned below:

a) Swap Homes (S,T_i,T_j)

This is the first step to make a change in the initial schedule. What we have to do is the schedule which we obtained after applying approach to BBO we need to swap home locations of team T_i,T_j together to make an upgradable in obtained schedule.

Table 2 Initial schedule:

T/R	1	2	3	4	5	6	7	8	9	10
1	6	-2	4	3	-5	-4	-3	5	2	-6
2	5	1	-3	-6	4	3	6	-4	-1	-5
3	-4	5	2	-1	6	-2	1	-6	-5	4
4	3	6	-1	-5	-2	1	5	2	-6	-3
5	-2	-3	6	-4	1	-6	-4	-1	3	2
6	-1	-4	-5	2	-3	5	-2	3	4	1

Flow Chart of ESA

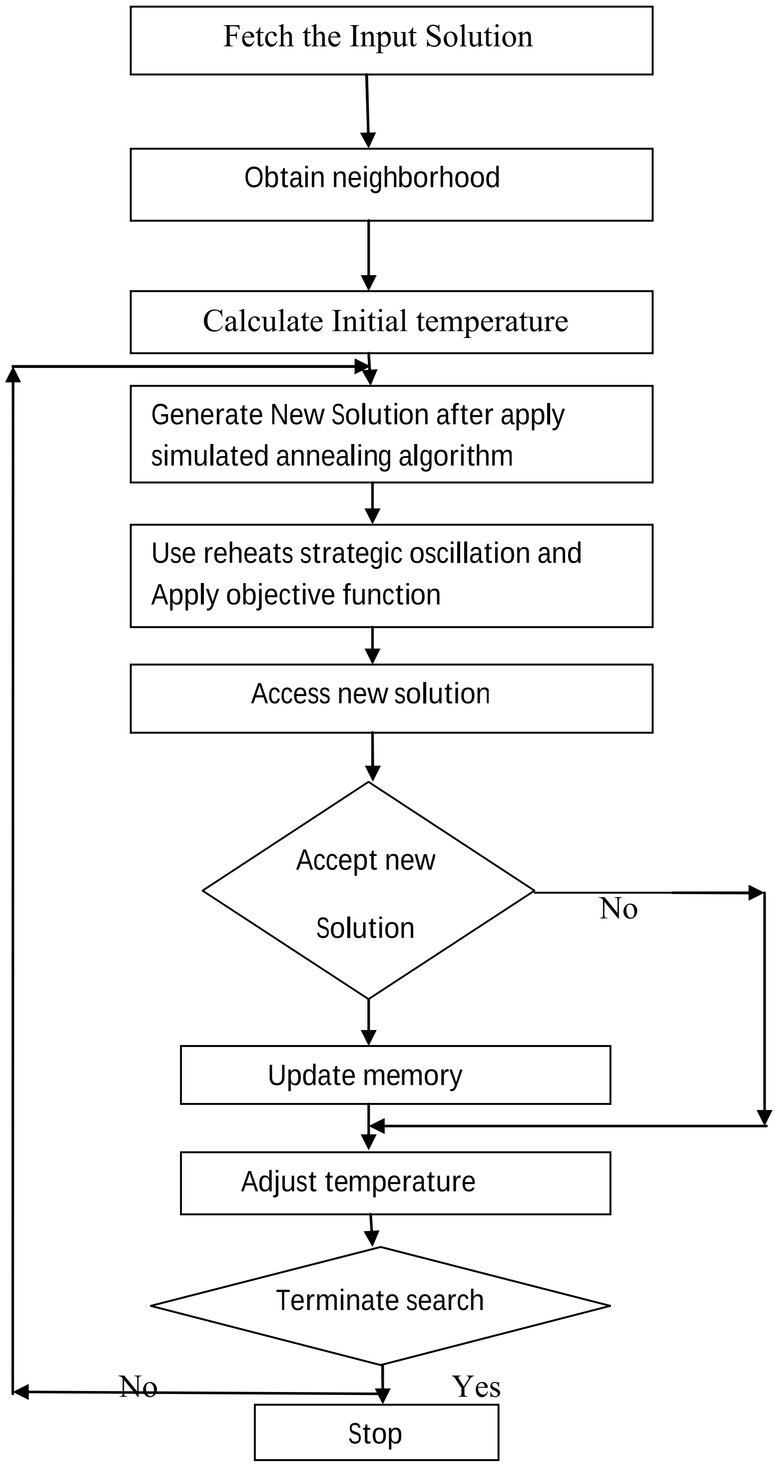


Fig 6 Flow chart illustrating working of ESA

After swapping homes it gets converted into

Table 3 Schedule produced after swapping of homes

T/R	1	2	3	4	5	6	7	8	9	10
1	6	-2	4	3	-5	-4	-3	5	2	-6
2	5	1	-3	-6	-4	3	6	4	-1	-5
3	-4	5	2	-1	6	-2	1	-6	-5	4
4	3	6	-1	-5	2	1	5	-2	-6	-3
5	-2	-3	6	-4	1	-6	-4	-1	3	2
6	-1	-4	-5	2	-3	5	-2	3	4	1

b) Swap rounds (S,□□□□□

Here we swap rounds. These are the □□□□ two rounds which we need to swap. Here we need to use O□□□) moves to proceed further.

Table 4 Input to swap rounds

T/R	1	2	3	4	5	6	7	8	9	10
1	6	-2	4	3	-5	-4	-3	5	2	-6
2	5	1	-3	-6	-4	3	6	4	-1	-5
3	-4	5	2	-1	6	-2	1	-6	-5	4
4	3	6	-1	-5	2	1	5	-2	-6	-3
5	-2	-3	6	-4	1	-6	-4	-1	3	2
6	-1	-4	-5	2	-3	5	-2	3	4	1

Obtained schedule after implementation of above move

Table 5 Schedule produced after applying of swap rounds

T/R	1	2	3	4	5	6	7	8	9	10
1	6	-2	-5	3	4	-4	-3	5	2	-6
2	5	1	-4	-6	-3	3	6	4	-1	-5
3	-4	5	6	-1	2	-2	1	-6	-5	4
4	3	6	2	-5	-1	1	5	-2	-6	-3
5	-2	-3	1	4	6	-6	-4	-1	3	2
6	-1	-4	-3	2	-5	5	-2	3	4	1

c) Swap teams (S, ▯▯▯▯▯▯

Swap team ▯▯▯▯▯.

Table 6 Input to swap team move

T/R	1	2	3	4	5	6	7	8	9	10
1	6	-2	4	3	-5	-4	-3	5	2	-6
2	5	1	-3	-6	4	3	6	-4	-1	-5
3	-4	5	2	-1	6	-2	1	-6	-5	4
4	3	6	-1	-5	-2	1	5	2	-6	-3
5	-2	-3	6	4	1	-6	-4	-1	3	2
6	-1	-4	-5	2	-3	5	-2	3	4	1

Obtained schedule after implementation of above move

Table 7 Produced schedule after applying swap teams

T/R	1	2	3	4	5	6	7	8	9	10
1	6	-5	4	3	-2	-4	-3	2	5	-6
2	5	-3	6	4	1	-6	-4	-1	3	-5
3	-4	2	5	-1	6	-5	1	-6	-2	4

4	3	6	-1	-2	-5	1	2	5	-6	-3
5	-2	1	-3	-6	4	3	6	-4	-1	2
6	-1	-4	-2	5	-3	2	-5	3	4	1

d) Partial Swap rounds (S,□□□, □□□, □

For the given team □□ we need to swap □□□□ partially i.e. only few values of a selected round gets changed

Table 8 Input to partial swap rounds

T/R	1	2	3	4	5	6	7	8	9	10
1	6	-2	2	3	-5	-4	-3	5	4	-6
2	5	1	-1	-5	4	3	6	-4	-6	-3
3	-4	5	4	-1	6	-2	1	-6	-5	2
4	3	6	-3	-6	-2	1	5	2	-1	-5
5	-2	-3	6	2	1	-6	-4	-1	3	4
6	-1	-4	-5	4	-3	5	-2	3	2	1

Obtained schedule after implementation of above move

Table 9 Schedule produced after partial swap rounds

T/R	1	2	3	4	5	6	7	8	9	10
1	6	4	2	3	-5	-4	-3	5	-2	-6
2	5	-6	-1	-5	4	3	6	-4	1	-3
3	-4	5	4	-1	6	-2	1	-6	-5	2
4	3	-1	-3	-6	-2	1	5	2	6	-5
5	-2	-3	6	2	1	-6	-4	-1	3	4
6	-1	2	-5	4	-3	5	-2	3	-4	1

.

e) Partial Swap Teams (S,[illegible]

Table 10 Input to partial Swap teams

T/R	1	2	3	4	5	6	7	8	9	10
1	6	-2	4	3	-5	-4	-3	5	2	-6
2	5	1	-3	-6	4	3	6	-4	-1	-3
3	-4	5	2	-1	6	-2	1	-6	-5	2
4	3	6	-1	-5	-2	1	5	2	-6	-5
5	-2	-3	6	4	1	-6	-4	-1	3	4
6	-1	-4	-5	2	-3	5	-2	3	4	1

Optimal neighborhood

Table 11 Schedule produced after partial swap teams

T/R	1	2	3	4	5	6	7	8	9	10
1	6	-2	2	3	-5	-4	-3	5	4	-6
2	5	1	-1	-5	4	3	6	-4	-6	-3
3	-4	5	4	-1	6	-2	1	-6	-5	2
4	3	6	-3	-6	-2	1	5	2	-1	-5
5	-2	-3	6	2	1	-6	-4	-1	3	4
6	-1	-4	-5	4	-3	5	-2	3	2	1

We make the initial schedule represented in table 1 which is produced after roulette wheel selection method on a random permutation of 1 to 6 numbers. We apply roulette wheel selection method five times to generate the first half of the schedule. Applying the above swapping methods to our initial schedule we generate the neighborhood. This is the last swapping that we did to our schedules.

4.1.3.2 Simulated Annealing Algorithm:

Now we apply simulated annealing algorithm to the above neighborhood in order to converge our solution.. Previously it has been done in three steps as discussed in [7], but by doing it in five steps and collaborate it with all extended abundance species models we are able to produce better results. Simulated annealing algorithm used is:

1. find random schedule S;
2. bestSoFar ← cost (S); counter ← 0;

PK [illegible] ≤ [illegible] m [illegible]

QK phase ← 0;

RK counter ← 0;

SK [illegible] ≤ [illegible] ` [illegible]

TK [illegible] Ep F X

UK [illegible] p′ [illegible] p [illegible] X

VK [illegible] Ep′F Y [illegible] Ep F [illegible]

NMK [illegible] ← [illegible] X

NNK [illegible]

NOK [illegible] ← [illegible] E | Lq H

NPK [illegible] X

NQK [illegible]

NRK [illegible]

NSK p ← p′ X

NTK [illegible] p′F Y [illegible] p c [illegible]

NUK counter ← 0; phase ← 0;

NVK [illegible] p c ← [illegible] p′F X

OMK [illegible]

ONK [illegible] HHX

OOK [illegible]

OPK [illegible]

OQK [illegible]

ORK [illegible] HHX

OSK T ← T · _;

OTK [illegible]

4.1.3.3 Strategic Oscillation

[illegible]

4.1.3.4 Reheats

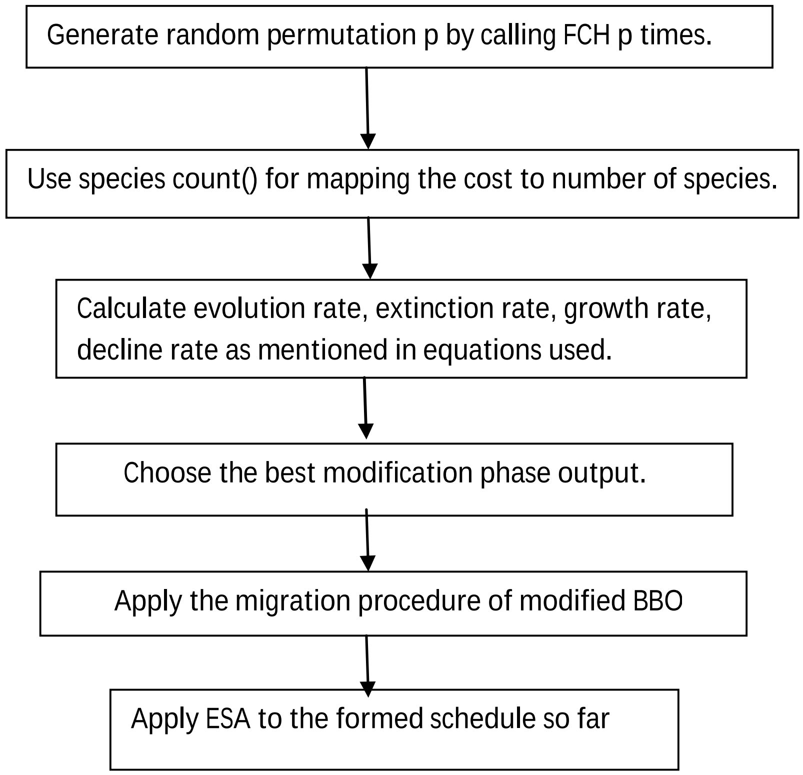
[illegible]

4.1.3.5 Objective function

1. [illegible]
2. BestFeasible ← ∞;
3. NBF ← ∞;
4. BestInfeasible ← ∞;
5. NBI ← ∞;
6. Reheat ← 0;
7. Counter_var ← 0;
8. [illegible]
9. Phase_var ← 0;
10. [illegible]
11. Counter_var ← 0;
12. [illegible]
13. [illegible]
14. [illegible]
15. [illegible]
16. [illegible]
17. [illegible]

```
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
NBF ← min(C(S), BestFeasible);
[illegible]
NBI ← min(C(S), bestInfeasible);
[illegible]
[illegible]
reheat_var ← 0; counter_var ← 0; phase ← 0;
BestTemperature ← T ;
BestFeasible ← NBF ;
BestInfeasible ← NBI;
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
T ← 2 · BestTemperature;
[illegible]
```

4.2 Functional architecture of proposed algorithm



[illegible]

4.3 Cost function used

[illegible]

[illegible]

4.4 Fitness Function Used

[illegible]

[illegible]

[illegible]

4.5 Proposed Algorithm

[illegible]

NK [illegible]

OK [illegible]

PK **for** [illegible]

QK [illegible]

RK [illegible]

SK **end for**

c [illegible] TKN [illegible]

NK t [illegible] Kc [illegible] c ` e [illegible] K

OK t [illegible] e pf [illegible] K

PK t [illegible] K

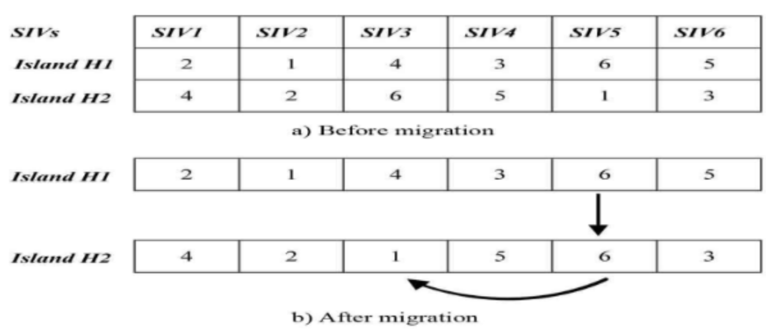
QK t [illegible] Kj [illegible]

RK t [illegible] Ki [illegible] W

SK t [illegible]

TK r [illegible] Km [illegible] K

UK t [illegible] k i S [illegible] SK


SIVs
SIV1
SIV2
SIV3
SIV4
SIV5
SIV6
Island H1
2
1
4
3
6
5
Island H2
4
2
6
5
1
3
a) Before migration
Island H1
2
1
4
3
6
5
Island H2
4
2
1
5
6
3
b) After migration

Chapter 5

COMPUTATIONAL RESULTS

[illegible]

5.1 Parameter Specifications

[illegible]

5.2 Matlab Simulation over 100 generations

[illegible]

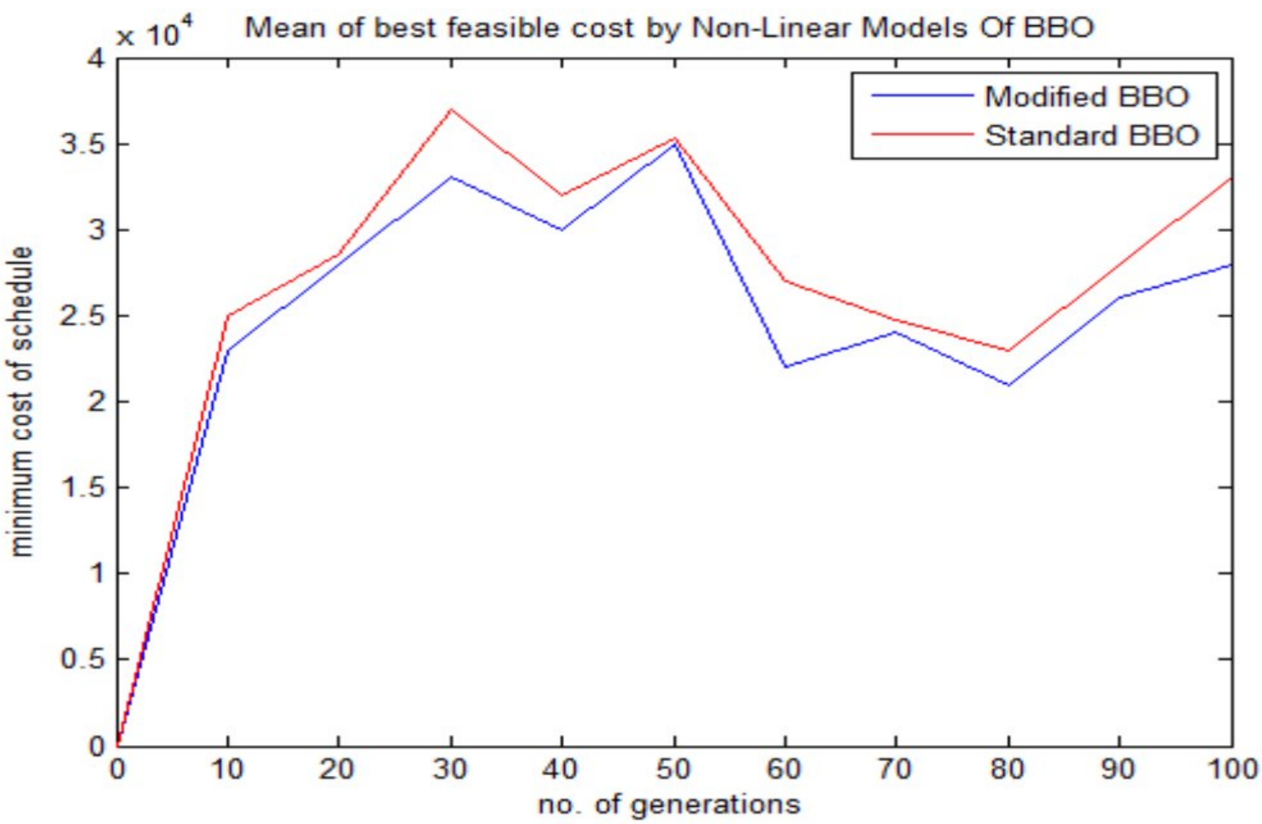
5.2.1 Linear Model comparison

- [illegible]

q □□□=NOo □□□□=□= □□□□=__l =□□=bp □=□□=□□□□= □□□□

f□□□□□	_ □□=h □□□□	__l Ip□	j □□□□□ __l Ibp□	d □Ip□	a □□□□□□= □□□=__l
k i =S	OUNOO	OUUMO	OUTRT	OUVVM	QR
k i =U	QNVOU	QOUMR	QOTUV	QPNNO	NS
k i =NM	SPUPO	SSPPN	SSPPQ	SSOSQ	JP
k i =NO	NNVSMU	NONMTM	NONMNN	NOMVUN	RV
k i =NQ	NVVPSP	ONMNPO	ONMNMQ	OMUMS	OU
k i =NS	OTUPMR	OVNPVQ	OVNPNM	OVMNUU	UQ

q □=□□□□=□□□□=□□□□□□□=□=□ □□□ □□=□□□=□□□=NMM=□□□□□□□□=□=□ □□□□=□□ □□□□□Kq □= □□□□□□□=□□=□□□□□□□=□□ □□□=□=□□□□□□□□□=□□=□□□□□=□□=□□□□□□□□=□=□ □□□ □□=□□□= □=□□□□□□=□□□□□□K



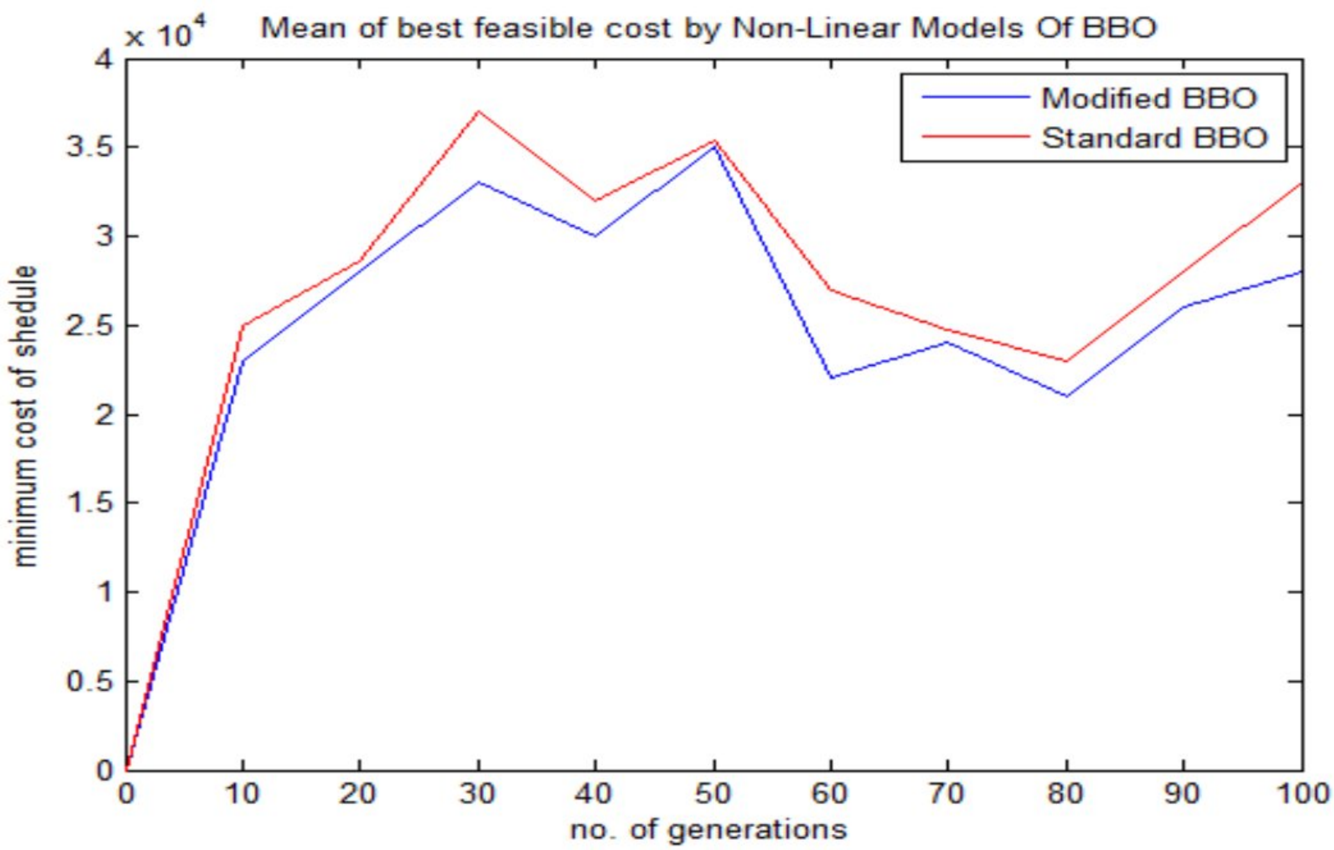
c□V` □□ □□□□□□=□□□□=□□□□□□=□=□□□□= □□□=□=j □□□□□=__l =□□=□p □□□□□=__l

5.2.2 Non Linear Model comparison

- f

q NP o _ _ l bp

f	_ h	_ _ l Ip	j _ _ l Ibp	d Ip	a _ _ l
k i =S	OUNOO	OUUMO	OUTQO	OUVVM	SM
k i =U	QNVOU	QOUMR	QOTSQ	QPNNO	QN
k i =NM	SPUPO	SSPPN	SSPQO	SSOSQ	JNN
k i =NO	NNVSMU	NONMTM	NONMMR	NOMVUN	SR
k i =NQ	NVVPSP	ONMNPO	ONMMVS	OMUMUS	PS
k i =NS	OTUPMR	OVNPVQ	OVNOVQ	OVMNUU	NMM



c NM k c j _ _ l p _ _ l

[illegible]

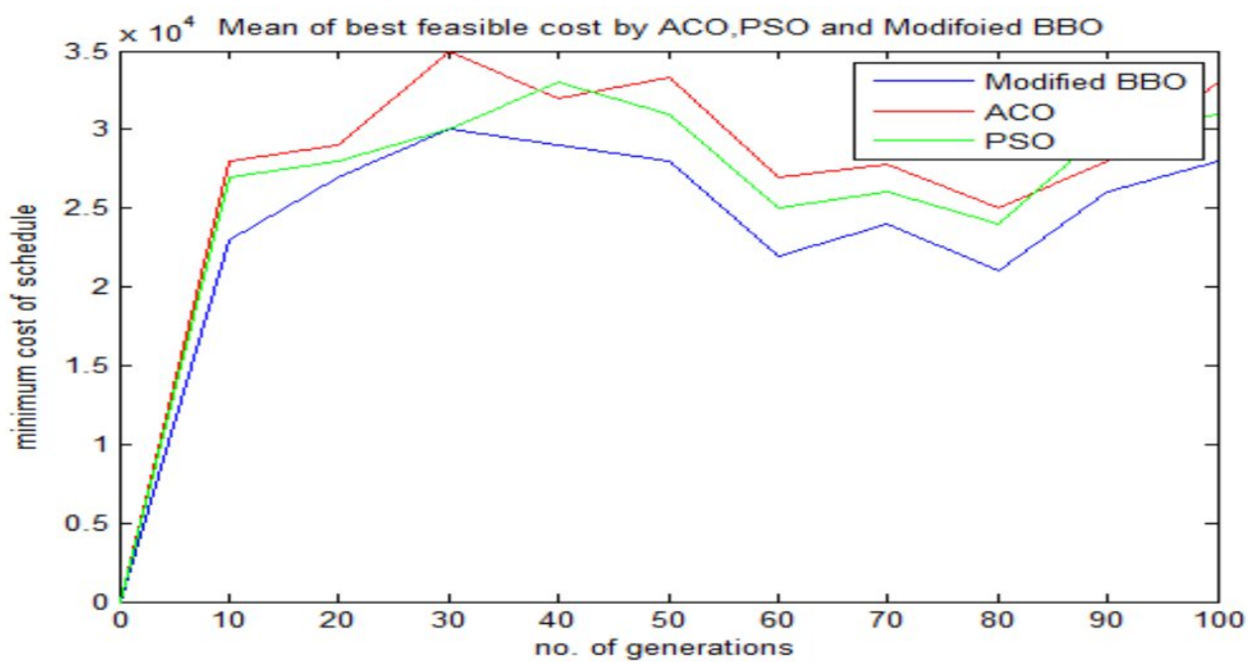
[illegible] Linear Models	BFC	BFS	BIC	BIS	Nov BF	Nov BI
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible] Non Linear Models						
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

5.3 Performance comparison with other Algorithms:

[illegible]

q [illegible] NRW
a [illegible] k i S [illegible]

	Team	ATL	NYM	PHI	MON	FLA	PIT
1	ATL	M	TQR	SSR	VOV	SMR	RON
2	NYM	TQR	M	UM	PPT	NMM	PNR
3	PHI	SSR	UM	M	PUM	NMOM	ORT
4	MON	VOV	PPT	PUM	M	NPUM	QMU
5	FLA	SMR	NMM	NMOM	NPUM	M	NMNM
6	PIT	RON	PNR	ORT	QMU	NMNM	M

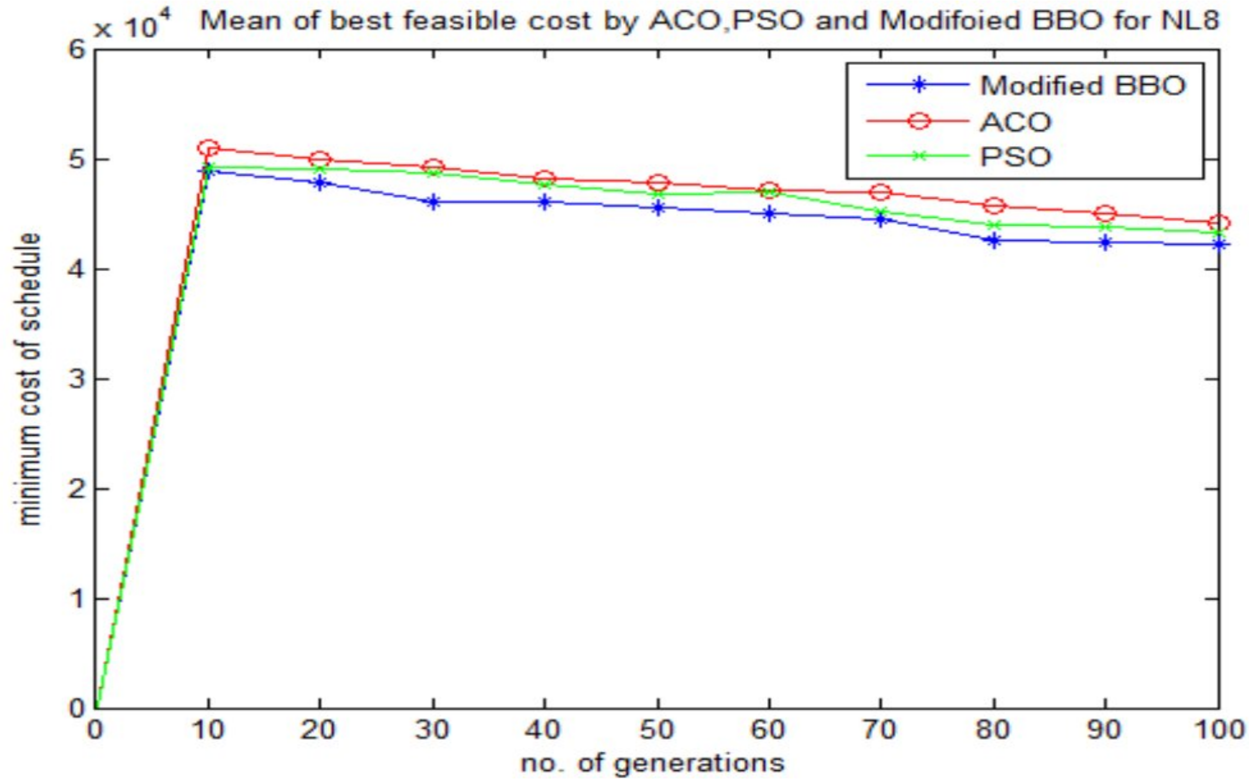


c [illegible] N` [illegible] _ [illegible] ` l l=mpl l=j [illegible] _ _ l [illegible] k i S q□□□□=NS=□□□ □□□□□□=□□□□□□□□=□□□□ □□□□=□□□

n	T_0	β	w_0	δ	θ	$maxC$	$maxP$	$maxR$	γ
8	400	0.9999	4000	1.04	1.04	5000	7100	10	2
10	400	0.9999	6000	1.04	1.04	5000	7100	10	2
12	600	0.9995	10000	1.03	1.03	4000	1385	50	1.6
14	600	0.9999	20000	1.03	1.03	4000	7100	30	1.8
16	700	0.9999	60000	1.05	1.05	10000	7100	50	2

q□□□□=NT a □□□□□□□=□ □□□□□=□□□=k i U=□□□□

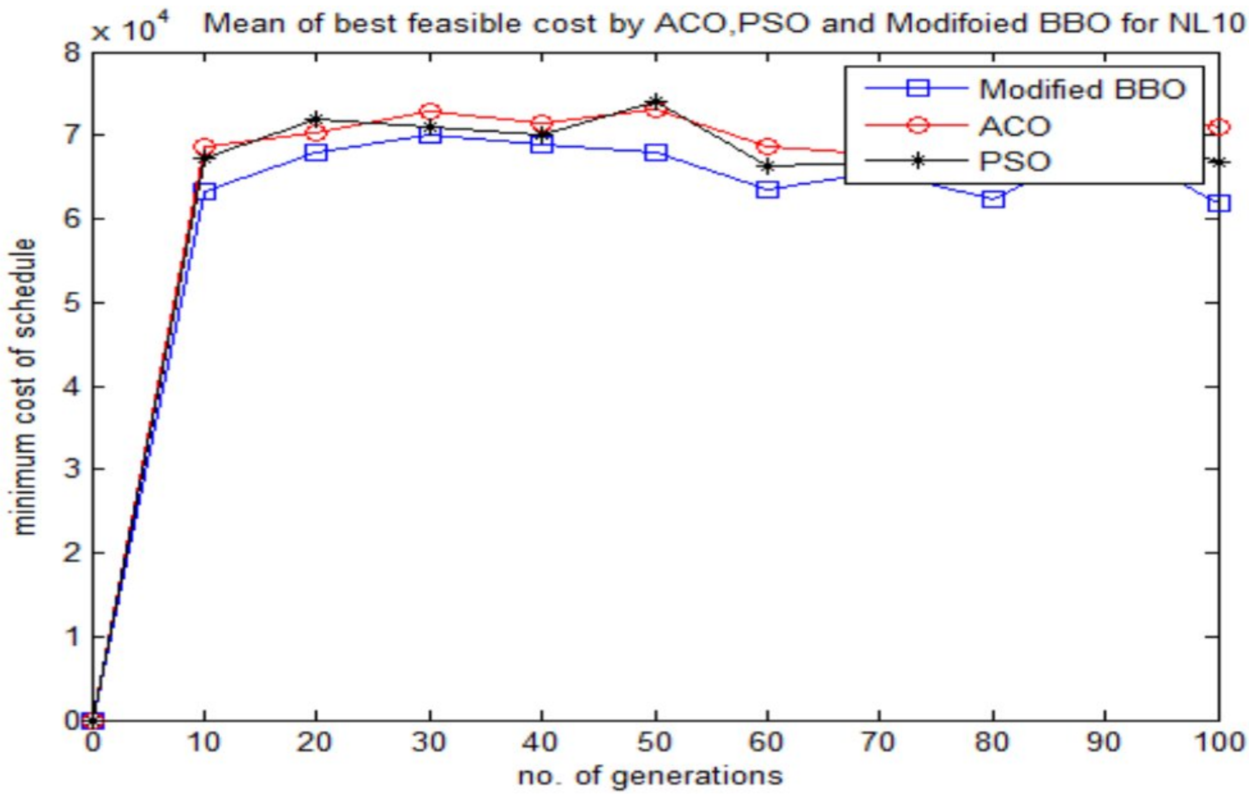
	1	2	3	4	5	6	7	8
1	0	745	665	929	605	521	370	587
2	745	0	80	337	1090	315	567	712
3	665	80	0	380	1020	257	501	664
4	929	337	380	0	1380	408	622	646
5	605	1090	1020	1380	0	1010	957	1190
6	521	315	257	408	1010	0	253	410
7	370	567	501	622	957	253	0	250
8	587	712	664	646	1190	410	250	0



[illegible]

[illegible]

	1	2	3	4	5	6	7	8	9	10
1	0	745	665	929	605	521	370	587	467	670
2	745	0	80	337	1090	315	567	712	871	741
3	665	80	0	380	1020	257	501	664	808	697
4	929	337	380	0	1380	408	622	646	878	732
5	605	1090	1020	1380	0	1010	957	1190	1060	1270
6	521	315	257	408	1010	0	253	410	557	451
7	370	567	501	622	957	253	0	250	311	325
8	587	712	664	646	1190	410	250	0	260	86
9	467	871	808	878	1060	557	311	260	0	328
10	670	741	697	732	1270	451	325	86	328	0


x 10^4 Mean of best feasible cost by ACO,PSO and Modifoied BBO for NL10
Modified BBO
ACO
PSO
minimum cost of schedule
no. of generations
0
1
2
3
4
5
6
7
8
0
10
20
30
40
50
60
70
80
90
100

[illegible]

[illegible]

Chapter 6

Conclusion and Future Scope

[illegible]

6.1 Conclusion

[illegible]

6.2 Future scope

Chapter 7

Publication from Thesis

Conference Name: [illegible]

Paper Title: [illegible]

Authors: [illegible]

Status: [illegible]

Location: [illegible]

Conference date: [illegible]

References

[1] D. [illegible] T. H. [illegible], [illegible] C. [illegible] O[illegible], [illegible] I[illegible], V[illegible] 23, [illegible] 33-38, 2005.

[2] F. B[illegible] L. [illegible], M[illegible] T[illegible] T[illegible] P[illegible]: [illegible] E[illegible], [illegible] *Advances in Artificial Intelligence - IBERAMIA-SBIA 2006*, [illegible] 4140, S[illegible], 208-217, [illegible] 208-217.

[3] P. C[illegible], G. K[illegible] G. B[illegible], [illegible] B[illegible] H[illegible]-[illegible] T[illegible] T[illegible] P[illegible], [illegible] *IEEE Symposium on Computational Intelligence in Scheduling*, H[illegible], HI, 2007.

[4] K. C[illegible], S[illegible] [illegible] D[illegible] O[illegible], [illegible] 15, [illegible] 139-142, 2008.

[5] F. [illegible], S. U. [illegible] C. R. [illegible] L S [illegible] 7 [illegible] I[illegible] [illegible] M[illegible], 2008.

[6] L. D. G[illegible], [illegible] S[illegible]-[illegible], J[illegible] H[illegible], [illegible] 163, [illegible] 189-207, 2007.

[7] J. D. E[illegible], E. L. [illegible] W. W. [illegible] S[illegible] [illegible] H[illegible] C[illegible] D[illegible], CRC P[illegible], [illegible] 578-584, 1995.

[8] K. E[illegible], T[illegible], [illegible] S[illegible], [illegible] PSO-S[illegible] T[illegible] T[illegible] P[illegible], [illegible] I[illegible] [illegible] & I[illegible] E[illegible], T[illegible], 2009.

[9] K. E[illegible], G. N[illegible], M. T[illegible], S[illegible] T[illegible] T[illegible] P[illegible]: [illegible] C[illegible] I[illegible] P[illegible] [illegible] R[illegible] [illegible], E. B. [illegible] P.D. C[illegible] (E[illegible]), P[illegible] T[illegible] [illegible] UIS, L[illegible] N[illegible] [illegible] C[illegible] S[illegible], [illegible] 2740 S[illegible], [illegible] [illegible], 2003.

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible] *2nd world Congr. on Nature and Biologically Insipired Computing* [illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible] *International conference on Computers & Industrial Engineering* [illegible]

[illegible]

[illegible] *European Journal of Operations Research,* [illegible]

[illegible] *IEEE symposium on computational intelligence in scheduling* [illegible]

[illegible] *The 2nd international conference on the practice and theory of automated timetabling, Lecture notes in computer science* [illegible]

[28] [illegible]

[29] [illegible] *International Joint Conference on on Computational Science and Optimization (CSO)*, [illegible]

[30] [illegible] *Theoritical computer science*, [illegible]

[31] [illegible]

Appendix A

[illegible]

1. Ackley

[illegible]

[illegible]

[illegible]

[illegible]

2. Quadratic

[illegible]

[illegible]

[illegible]

[illegible]

3. Sphere

[illegible]

[illegible]

[illegible]

[illegible]

Appendix B

Abbreviations

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]