Academic Curriculum Load Balancing for Students using Genetic Algorithms

Chakradhar M, Kunal M, Sri Charan M,Umesh Sai M, Y.V.S. Murthy, Shashidhar G.K.,
Deapartment of CSE, National Institute of Technology Karnataka,
Surathkal, Mangalore, India - 575 025.
chakradharmalagumdla@gmail.com, sreecharanmallu04@gmail.com, saiumesh333@gmail.com,urvishnu@gmail.com, koolagudi@nitk.edu.in

***Abstract*—In the paper, we propose an algorithm using genetic alogithm to find out the optimal solution for the academic load balancing problem. The load balancing problem is to optimize the load of credits per semester in an academic curriculum. In the proposed method, we try to distribute the course load as evenly as possible so that the deviation from the mean credit load per each semester is as minimal as possible. The objective function is to distribute the credit load among all the semesters evenly such that the deviation from the mean credits per semester is minimal. The proposed approach explores the solution space using only mutation operators and does not operate using crossover as the solutions obtained using cross over does not create any newer and better solutions in the solution space.The algorithm is applied on three data sets and the results are compared with the solutions obtained using the existing approaches. The results obtained using the state of the art solution are either better than approaches or on par with the state of art optimal solutions. The solution set obtained using the proposed approach is well spread out through out all the periods and all the periods contain almost mean number of credits.**

***Index Terms*—Alternating frequency, BACP, Fitness function, Maximum generations, Objective function, Population size, Shift intensity, Shift operator,Swap operator, Tournament selection.**

I. Introduction

In most of the colleges, the academic curriculum is divided into semesters. It usually happens that certain semesters are more hectic and crowded and some semesters are comparatively free in terms of the effective academic load of that semester. This makes it difficult for a student learn all the courses effectively and compete with others.The main problem faced by any institute in this regard is to find out an optimal solution such that load in each semester is well-balanced.So, the problem is to balance the academic load of each semester in a curriculum.The problem becomes tougher as certain courses have prerequisites and each semester has course and credit constraints.In this paper, we present an approach based on mutation only genetic algorithm to optimize the solution to the BACP. In many colleges the courses are taken by the students in every semester. The courses in every semester are to be balanced in order to limit the workload of the students and there also exists prerequisites such that some courses can be taken only after completing others(for example Advanced data structures can be taken only after completing Data structures). This problem in the colleges is called Balancing Academic Curriculum Problem.

The Balancing is helpful to students in finding the optimally balanced semesters with courses given to them.

The BACP problem is implemented using genetic algorithm with swap and shift mutation operators which shift and swap courses across semesters while searching the solution space.

The remaining of this paper is organized as in the following. Section II presents a literature review of approaches solving the BACP problem. Section III presents proposed approach. Section IV presents the results obtained and observations based on results. Section V concludes the paper.

II. Literature Review

Several proposals are made for solving the Balancing Academic Curriculum problem. [1] Proposed the method for BACP based on Real Life Instances.Heuristics have been designed, implemented and analysed basd on local search based on professors preference and based on these such real-life problems a new objective function is developed.This proposed model does not distribute the courses evenly along the semesters as this is done based on local search. .

[2] presented experimental results of solving mathematical models, using different techniques, for designing balanced academic curriculum. So this also would give result in such a way that academic period of the curriculum have equally distributed load. Based on this work, a software has been developed for this academic curriculum.[2] solved the problem using constraint Programming.It is a mathematical model which uses specific variables and conditions. they presented experimental results of solving mathematical models, using different techniques, for designing balanced academic curriculum. So this also would give result in such a way that academic period of the curriculum have equally distributed load. Based on this work, a software has been developed for this academic curriculum.This technique decreases with the accuracy when the iterations keep increasing.

[3] proposes the algorithm based on the hybrid resolution framework which also includes genetic algorithms for solving the BACP. Many resolution algorithms have been proposed and can be classied in two main groups. Complete methods and Incomplete methods.So in the complete method the whole research has been done to find that the constraint programming is a inconsistent. . It is a twofold problem with constraint

programming techniques and also genetic modelling so that it manages new and finer solving strategies and extensions.This solves the problem using several elementary functions. This adapts to the new techniques and properties can be extended but this model is complex and takes relatively much time in producing the efficient solution.

So they have done the split of domains for constraint programming which increased their accuracy.But Inconsistent methods result in use of meta-heuristics also provides more efficient solution .But in this paper we would rather discuss on fitness function and Genetic algorithms. This type of technique solves the problem by using several elementary functions.

[4] solves the BACP by using 2 different objective functions based on the mathematical models.They are Integer programming and constraint programming.These are NP hard problems.when problems become too hard, these NP problems cannot be solved using complete methods. Assuming that an incomplete approach can obtain good solutions very quickly, they have solved a real-life combinational problem using a Genetic Local Search algorithm .A local search genetic algorithm has been designed.When compared to complete method it is better but it focuses more on realistic solution rather than optimised solution.Assumes that NP problems cannot be solved using complete methods and solves the problem incomplete approach using Genetic Local Search algorithm.This approach helps in solving the problem without using constraint programming techniques. A local search genetic algorithm has been designed.When compared to complete method it is better but it focuses more on realistic solution rather than optimised solution.

[5] solves the problem of BACP using Tabu search.Tabu search is a method used to solve combinatorial optimization problems. The main idea behind the Tabu search method is that by using a memory forces the method to explore new areas in the search space. That is, it can memorize some solutions that have been examined recently and these points become forbidden (taboo) to make decisions about the following solution. This method is used in the following structure to represent the different actors involved in the model. To keep track of movements that have been made the authors use a two dimensional array of size total number of periods (ntp).fitness function used is the sum of the absolute error.Thus it Proposes the algorithm by using some memory storing strategies and help in finding the solution for all the periods based on these.But this requires an extra memory for generating the solution.

[6] Proposes algorithm is similar to proposed algorithm using genetic algorithm using mutation only operator where objective function is different in this case which is about finding the best individual based on the minimum of maximum credit load of all academic periods.This produces the result efficiently but it takes more iteration to find the optimal solution.

Our Proposed algorithm solves the BACP using genetic algorithm which uses a fitness functions that produces the optimal solution with relatively less number of iterations.

III. Proposed Methodology

A. Formulating BACP

Given below are the set of parameters for developing the solution for the problem.

Parameters:

m – Number of courses

n – Number of academic periods

α_i – Number of credits of course i, $\forall i=1, \ldots, m$

β – Minimum credit load allowed per period

γ – Maximum credit load allowed per period

δ – Minimum number of courses per period

ε – Maximum number of courses per period

Decision variables:

x_{ij} -equals 1, if course i is set in period j, otherwise it is 0, $\forall i=1, \ldots, m$, $\forall j=1, \ldots, n$

c_j – Credit load of period j, $\forall j=1, \ldots, n$

c – Maximum credit load of all periods

Objective function:

$$Min\ c = Max\{c_j\}, \text{ where } c_j = \sum_{i=1}^{n} \alpha_i x_{ij}, \quad \forall j=1, \ldots, n \tag{1}$$

Constraints:

$$\sum_{j=1}^{n} x_{ij} = 1, \forall i=1, \ldots, m \tag{2}$$

$$x_{bj} \leq \sum_{r=1}^{j-1} x_{ar} = 1, \forall j=2, \ldots, n \tag{3}$$

$$c_j \geq \beta, \forall j=1, \ldots, n \tag{4}$$

$$c_j \leq \gamma, \forall j=1, \ldots, n \tag{5}$$

$$\sum_{i=1}^{m} x_{ij} \geq \delta, \forall j=1, \ldots, n \tag{6}$$

$$\sum_{i=1}^{m} x_{ij} \leq \varepsilon, \forall j=1, \ldots, n \tag{7}$$

Constraints:

First equation (1) represents the objective function, It is minimising the deviation of course load from mean credits in all the periods (2)Constraint which checks all the courses are assigned to some period, (3) Checks all the prerequisite conditions are satisfied (4) Checks maximum load (5)Checks minimum load constraint. (6) checks the maximum courses for a period. (7) checks the minimum courses for a period

IV. Operators

These Swap and Shift Mutation operations are done based on [6] Swap operator swaps the courses between two different periods where as shift operator shifts the course from one period to other period.

sw is parameter which tells how many times swap operation has to be done between two different periods. In each iteration in these sw iterations two different period let say p and q are selected in a random fashion and also respective courses should be selected from both of them which are feasible by the meaning if they are swapped the prerequisite constraint should not be changed.So,This feasible swapping helps us in mutating the individual. If swap is not feasible then this swap operation is done for next iteration.

Algorithm 1 Swap operator procedure

```
Input: individual I, swap mutate intensity sw
Output: mutated individual I_m
I_m = I
i = 1
repeat
  p = Random period from I_m
  q = Random period from I_m different from p
  CL_p = Get courses that can be swept from p
  CL_q = Get courses that can be swept from q
  repeat
    I_m = Mutate I_m by randomly swapping a
    course from list CL_p with a course from list
    CL_q
  until a feasible swap cannot be found
  i = i+1
until i > sw
```

Shift operator shifts from one period to other period randomly if it is only feasible.This also is done for sh iteration where sh is parameter. So, in each iteration a random period is selected and a course is selected randomly in that period.So,in the remaining periods a this course is shifted to random period only if is feasible i.e., if it satisfies all the constraints and if this shift is not feasible then algorithm proceeds with next iteration.

Algorithm 2 Shift operator procedure

```
Input: individual I, shift mutate intensity sh
Output: mutated individual I_m
I_m = I
i = 1
repeat
  repeat
    p = Random period from I_m
    repeat
      c = Get a random course from p
    until no course provides a feasible shift
  until no period provides a feasible shift
  PL_q = Get periods that can accept course c
  if PL_q is not empty then
    q = Random period from PL_q
    I_m = Mutate I_m by shifting course c from
    period p to period q
  i = i+1
until i > sh
```

A. Initialisation

Academic curriculum balancing problem, courses are ordered based on the number of prerequisites they have, where courses with no prerequisites are placed first then courses with one prerequisite.

At the beginning , in order to start with a balanced curriculum, Initially all the periods contain same number of courses and if there are odd number of courses some periods will have extra courses. Next, by following the ascending order of the number of perquisites, the courses are dispatched into the available periods, subject to the constraints of min/max credit load and course perquisites. If during this phase, a given course cannot be settled at some particular period, due to the constraints, it gets shifted into one of the next available periods. As a result, the end of this phase will produce an individual that satisfies all the constraints. Further, in the second phase , using randomness intensity (is) a few randomness is done for initial individual by swapping between two courses between two periods that are selected randomly.

Algorithm 3 Initial solution procedure

```
Input: academic curricula C(m, n, α_i, β, γ, δ, ε), initial
solution randomness intensity is
Output: random individual I
Initialize periods of individual I to an empty state
IL = Get estimated number of courses for each
period
sp = 1
c = 1
repeat
  RC = Get required courses of course c
  p = sp
  repeat
    if number of courses in period p less then
    IL(c) then
      if course c satisfies the constraints of
      min/max credit load and none of the
      courses in RC is placed in period p
      then
        Add course c to period p of individual I
    else
      sp = sp + 1
    p = p + 1
  until p > n or c cannot be settled to period p
  c = c+1
until c > m
i = 1
repeat
  p = Random period
  q = Random period that is different from p
  I = Mutate I by randomly swapping a course
      from period p to period q
  i = i + 1
until i > is
```

V. Genetic Algorithm

Genetic Algorithm is continuous generative nature algorithm,it is a way of regenerating the population for every generation. There are 7 parameters that are used in this algorithm

1.population size (ps)
2.Total Generations(mg)
3.Tournament size(ts)
And operator parameters
4.Intensity randomness (is) for initialisation.
5.Swap mutation intensity(sw)
6.Shift mutation intensity(sh)
7.Alternating frequency parameter(af),which alternates between swap and shift.

After initialising the population the following steps are done in the algorithm

1. Finding out the best solution using the fitness function

2.Comparing the best solution with already found one using the fitness value of both solution.
3.Using the tournament selection algorithm selecting out the individual.
4.Performing Swap mutation or shift mutation based on Alternating frequency parameter.
5.Repeating the approach for mg iterations where mg is total number of generation that are there.

Algorithm 4 Genetic algorithm

Input: academic curricula $C(m, n, \alpha_i, \beta, \gamma, \delta, \varepsilon)$, population size *ps*, maximum generations *mg*, tournament size *ts*, initial solution randomness intensity *is*, swap mutate intensity *sw*, shift mutate intensity *sh*, operator alternation frequency *af*
Output: S_b

Operators = {Swap, Shift}
$P = \{\}$
$i = 1$
repeat
  $P = P \cup$ Random Individual (C, *is*)
  $i = i+1$
until $i > ps$
S_b = Best solution from P
$k = 1$
$j = 1$
repeat
  S_c = Best solution from P
  if S_c better S_b **then**
    $S_b = S_c$
  $Q = \{\}$
  $i = 1$
  repeat
    I = Select Individual (P, *ts*)
    Mutate = Operators (k)
    $Q = Q \cup$ Apply Mutate (I, *sw*, *sh*)
    $i = i+1$
  until i $> ps$
  $P = Q$
  $k = (i / af)\ \%\ 2$
  $j = j+1$
until j $> mg$

A. Fitness function for evaluating the best individual

The Fitness function finds out fitness value for each individual in the population.For a curriculum to be rightly balanced credit load should be divided equally among all the semesters so that the student work load becomes balanced.So, the fitness function evaluates every individual and produces fitness value for each one based on how equally the load is distributes and selects the best suitable one.

B. Set of solutions

A student may not always want the curriculum with best solution to be maximum credits as least as possible for a period sometimes student can find better curriculum in the other solutions so the best 10 solutions with curriculum is produced to student so that he can choose among the best options he has.

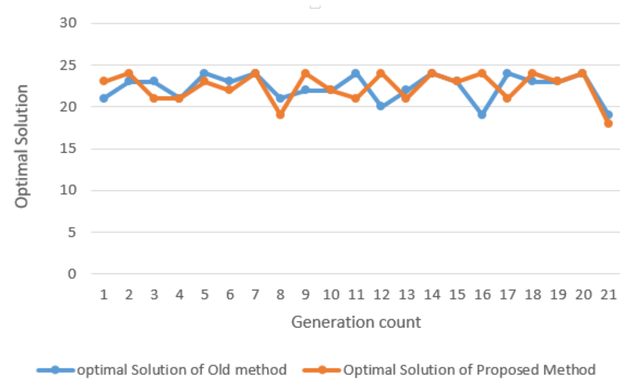

Fig. 1. Comparison for BACP-8 Data set.

VI. Experiments

In this section, firstly the test used for experimentation is described and then the results for algorithm parameters and also the comparing results with the optimal solutions of known approach. The algorithm is developed using Python programming language. Hardware specifications of the test environment used are CPU with i5 processor and 8GB of RAM memory and Windows10 Operating System.

A. Test Set

Proposed algorithm is evaluated for 3 data sets found on CSPLib.So the first dataset contains 8 periods, 46 courses, 133 credits and 4 maximum prerequisites per course.Second contains 10 periods, 42 courses, 134 credits and 3 maximum prerequisites per course.Third data set contains contains 12 periods, 66 courses, 204 credits and 5 maximum prerequisites per course. newline

Table I Details of test instances

Instance	Number of periods	Number of courses	Total credits	Maximal number of prerequisites per course
BACP8	8	46	133	4
BACP10	10	42	134	3
BACP12	12	66	204	5

VII. Result and Observations

The below Graph is plot between optimal solution on y-axis and No. of generations considered on x-axis for BACP-8 data set. For proposed method, optimal solution is found in 8th generation which is pretty much early than the existing algorithm in which optimal solution is found in 17th generation. Also, optimal solution of the already existing solution is not as well-spread as the proposed solution.This graph indicates that proposed algorithm gives more optimal solution in a less number of iterations.

The comparison results with BACP-8 data set have been depicted in Figure 1.

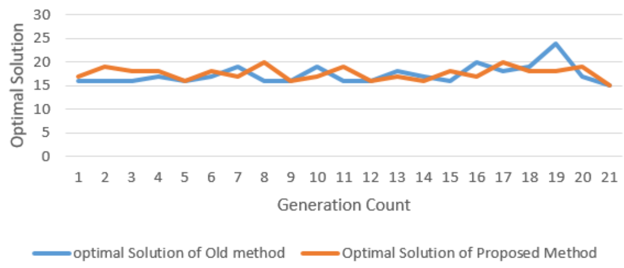

Fig. 2. Comparison for BACP-10 Data set.

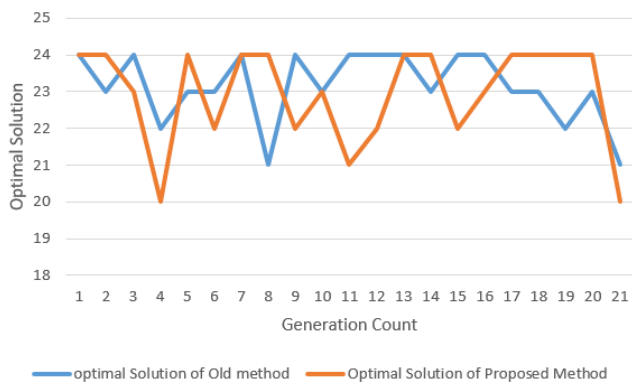

Fig. 3. Comparison for BACP-12 Data set.

The graph is a plot for BACP-10 data set.Here optimal solution of the state of art solution is same as that of existing algorithm but the deviation from optimal solution is very much less when compared with the reference algorithm.

Plotting for BACP-12 is also done by comparing the results with the referenced algorithm.Here, clearly optimal solution is found in very less number of iterations when compared to reference algorithm and also the more optimal solution is found.

The figure 4 is a plot between fitness values on y-axis and number of iterations on x-axis.This shows that the fitness values are widely spread instead of concentrating at only one point which is highly beneficial for production more optimal solutions which will have highest fitness value. Above table is compares the optimal solutions for the state of art approach and already existing methods for different data sets.Optimal solution here is the number of credits in the semester containing the maximum number of credits.

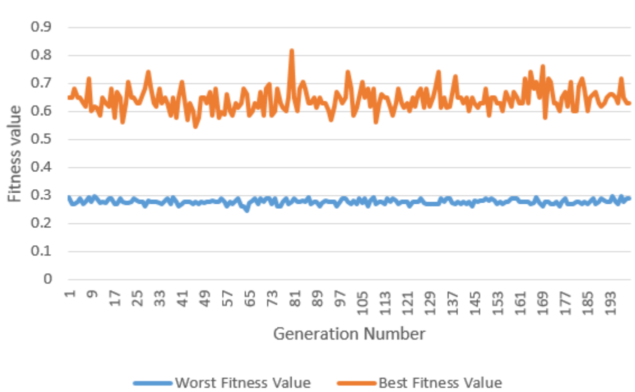

Fig. 4. Fitness value comparision.

Data set\Optimal Solution	State of the art approach	Existing Approach
BACP-8	18	19
BACP-10	15	15
BACP-12	20	21

VIII. Conclusion and Future work

In this paper, we present a modified approach for balancing the academic curricula problem based on genetic algorithm. We used only mutation operators for creating new off springs. The two operators swap and shift are used alternatively and newer solutions are created. The shift operator exchanges two courses in different periods such that the load constraints are satisfied and the shift operator shifts the course from one period to another to create a new individual in the population.

The results show that the algorithm provides solutions that are either on par or better than the solutions obtained by using the state-of-art method. Also, the results shoe that the optimal solution can be achieved in the method proposed with lesser number of generations indicating that the time taken to obtain the optimal solution is lesser and hence it is comparatively effective.

References

[1] M. Chiarandini, L. Di Gaspero, S. Gualandi and A. Schaerf, ”The balanced academic curriculum problem revisited,” Journal of Heuristics, vol. 18, no. 1, pp. 119-148, 2012.

[2] C. Castro and S. Manzano, ”Variable and value ordering when solving balanced academic curriculum problems,” in Proceedings of the ERCIM Working Group on Constraints, Prague, Czech Republic, 2001.

[3] T. Lambert, C. Castro, E. Monfroy and F. Saubion, ”Solving the balanced academic curriculum problem with an hybridization of genetic algorithm and constraint propagation,” in International Conference on Artificial Intelligence and Soft Computing, Zakopane, 2006.

[4] C. Castro, B. Crawford and E. Monfroy, ”A genetic local search algorithm for the multiple optimisation of the balanced academic curriculum problem,” in Cutting-Edge Research Topics on Multiple Criteria Decision Making, Springer Berlin Heidelberg, 2009, pp. 824-832.

[5] L. V. Rosas-Tllez, J. L. Martnez-Flores and V. Zanella-Palacios, ”Solution to the Balanced Academic Curriculum Problem Using Tabu Search,” Computer Technology and Application, vol. 3, no. 9, pp. 630-635, 2012.

[6] Kadri Sylejmani, Arbnor Halili, Arbnor Rexhepi ”Mutation

only genetic algorithm".,MIPRO 2017, May 22- 26, 2017, Opatija, Croatia