CHAPTER 3
GENETIC FLOWER POLLINATION ALGORITHM FOR RSMCDD SCHEDULING

Objective : To develop a nature-inspired evolutionary sequential hybrid metaheuristics namely Genetic Flower Pollination Algorithm (GFPA) to tackle restricted single machine common due date scheduling problem. Genetic algorithms are not suitable to solve larger job size problems as they take more computation time (nearly three days) to generate high quality solutions. The purpose of developing a hybrid metaheuristic algorithm is to provide high quality solutions in an acceptable computation time while solving larger job instances. The steps involved in developing hybrid metaheuristic algorithm have been discussed in detail in this chapter.

3.1 GENETIC ALGORITHM

Genetic algorithms are global search algorithms based on the evolutionary concepts of natural selection and genetics. These evolutionary algorithms follow Charles Darwin’s “Survival of the Fittest” principle. GA mimics natural selection thus evolving from a population of chromosomes to a new population by natural evolutionary genetic operators such as reproduction. Reproduction is achieved through genetic operators namely crossover and mutation.

Genetic algorithm starts with a set of solutions represented by chromosomes called population. After an initial population is randomly generated, the algorithm undergoes selection, crossover and mutation. Genetic algorithm maintains a population of chromosomes with fitness values associated
with it. Selection follows survival of the fittest strategy in which high quality solutions stay within the population. Selected parents undergo crossover in the mating pool to produce a new offspring. The best chromosomes have more chances to reproduce in the subsequent generations. The purpose of mutation is to maintain diversity within the population and inhibit premature convergence.

3.2 FLOWER POLLINATION ALGORITHM

Flower Pollination algorithm (FPA) (Xin-She Yang 2010; Yang 2012; Yang 2013) is a new nature-inspired metaheuristics based on the characteristics of flowering plants and viewed as a survival of the fittest optimization process for flowering plant species. This algorithm mimics the behavior of pollination taking place in flowering plants. Flower pollination algorithm has been applied to solve practical problems in engineering for single and multiobjective optimization problems.

The emergence of flowering plants happened over 160 million years ago on earth. 80% of the plant species are flowering plants. Most of the flowering plants are pollinated by animals. Some flowering plants are wind pollinated (anemophily) and water pollinated (hydrophily). The main purpose of a flower is reproduction which is achieved through pollination. Flower pollination is defined as the transfer of pollen and such transfer is associated with pollinators such as insects, birds, bats, animals, wind and water. Pollination can be achieved by either self-pollination or cross-pollination. Cross pollination is achieved when pollen of a flower from one plant is transferred to a pollen of a flower to another plant. Self-pollination is fertilization of one flower or different flowers of the same plant.

In flower pollination algorithm, pollination process occurs in two forms namely biotic and abiotic. 90% of the flowering plants belong to biotic
pollination as transfer of pollen is by pollinators such as insects, bats, birds and animals. Biotic cross pollination occurs at long distance and is named as global pollination. 10% of flowering plants do not require any pollinator and referred to as abiotic pollination. The two key steps involved in flower pollination algorithm are global pollination and local pollination.

3.3 PROPOSED GENETIC FLOWER POLLINATION ALGORITHM

When solving RSMCDD scheduling problem using GA, each solution is generally coded as a chromosome of finite length. Each chromosome is considered as an individual in the population. Permutation encoding scheme is used to code chromosomes. Population maintains a collection of chromosomes of size $2n$. The chromosomes in the initial population are generated by constructive heuristics that satisfy the three scheduling optimality properties. The offsprings are generated from the current population by applying roulette wheel selection strategy, ordered crossover and random swap followed by sliding mutation. The best individuals obtained are preserved. The quality of each chromosome is measured by a fitness function using Equation (1.5) and the search process proceeds until the termination criteria is met. Maximal number of generations is defined as the stopping criteria. Chromosomes with minimum fitness values are retained by adopting elitism strategy. The drawback in this approach is that GA takes three days to generate quality solutions for larger job size of 1000 resulting in greater computation time.

It is inferred that GAs are not suitable for solving larger job size scheduling problems thereby paving way to the development of hybrid metaheuristics to provide high quality solutions in an acceptable time. Nature inspired algorithms have accomplished appreciable success in the field of combinatorial optimization.
The proposed work focuses on the development of a new nature-inspired evolutionary hybrid metaheuristic algorithm namely Genetic Flower Pollination Algorithm (GFPA) to solve RSMCDD scheduling problems. The proposed algorithm gives scope to model GA based on a new Flower pollination algorithm thereby resulting in Genetic Flower Pollination Algorithm.

In order to reduce the computation time for larger jobs of single machine common due date job scheduling problem, the evolutionary characteristics of GA can be hybridized along with the pollination strategies of flower algorithm. Until now, nature-inspired flower pollination algorithm has not been explored in the realm of RSMCDD scheduling problem. Hence the proposed work attempts to hybridize genetic algorithm based on flower pollination algorithm.

### 3.4 STEPS INVOLVED IN PROPOSED GFPA

Genetic flower pollination algorithm, a new population-based nature-inspired evolutionary metaheuristics, modeled based on the inspiration from flowers is developed to solve RSMCDD scheduling problem. In this research work, the selection and reproduction strategies of GA are evolved based on the characteristics of pollination process and behavior of pollination (Yang 2012; Yang 2013) in flowering plants.

Each solution in GFPA is coded as a flower with a finite length. Each flower is considered as an individual. A collection of flowers of size 2*n forms a population where n denotes the size of the job. The flowers of the initial population are generated randomly that satisfy the three scheduling optimality properties. New offsprings are generated from the current population by pollination strategy. The quality of each flower is measured by a fitness function.
Algorithm terminates automatically when the flowers in the population are said to have same fitness values.

The proposed metaheuristics GFPA formulates three rules for RSMCDD problems similar to the rules (Yang 2012; Yang 2013) cited in flower pollination algorithm.

Rule 1) Crossover is considered as a process of global pollination.

Rule 2) Mutation is considered as local pollination process.

Rule 3) The interaction between flowers is determined by a switch probability $P_c \in [0..1]$ in pollination strategy.

3.4.1 Pollen Representation

Permutation encoding is the most suitable encoding mechanism for job scheduling problems. RSMCDD scheduling problem uses permutation encoding scheme to represent job sequences where each job is denoted as a pollen. A job sequence is mapped into a pollen gamete / flower with the pollen assuming different and non-negative integer values in the [1..n] interval. For a 10 jobs problem, the complete flower / job sequence is represented as ([1][2][3][4][5][6][7][8][9][10]) where [i] is the position of the $i^{th}$ job in the sequence. An example for permutation encoding representation for flowers is given in Table 3.1.

<table>
<thead>
<tr>
<th>Table 3.1 Permutation encoding representation for flowers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flower 1 : 4  2  1  3  10  7  9  6  5  8</td>
</tr>
<tr>
<td>Flower 2 : 4  2  3  7  6  9  5  8  1  10</td>
</tr>
</tbody>
</table>
3.4.2 Initial Population Construction

In RSMCDD scheduling problem, job of size ‘n’ is divided into two sets: n/2 jobs in early set and n/2 jobs in tardy set. The jobs that complete their execution before the common due date are stored in a set called $S_E$ and jobs that complete their execution after the common due date are stored in a set called $S_T$. The proposed hybrid metaheuristics generates $2 \times n$ feasible job sequences that satisfy the three scheduling optimality properties mentioned in Chapter 1.

To satisfy optimality property 3 in the population, instead of starting the first job at time 0, idle time is inserted at the beginning of the flowers so that the last job of $S_E$ coincides with the due date. Hence the proposed hybrid metaheuristics determines the two sets $S_E$ and $S_T$, constructs a sequence for each set and builds the final schedule by combining these two sets. To satisfy optimality property 1, no idle time is inserted between consecutive jobs. To satisfy optimality property 2, jobs in $S_E$ and $S_T$ are sequenced according to "\" shaped and "/" shaped respectively.

Instead of using an enumerative approach of generating $n!$ possible sequences of job size $n$, only feasible job sequences of size $2n$ are explored. Feasible job sequences are the job sequences that satisfy the three scheduling optimality properties of common due date scheduling problems.

3.4.3 Evaluation of Fitness Function

All pollen flowers in the population are evaluated using the fitness function as defined in Equation (1.5) and the pollen flower or job sequence with minimum fitness value is chosen as $g_{best}$ for that generation. Hence $g_{best}$ changes after every generation.
3.4.4 Determination of Switch Probability

Pollination activities can be carried out globally as well as locally to determine the interaction taking place between flowers in the population by satisfying rule 3. The interaction between flowers is determined by a control parameter called switch probability $P_c$. To determine the type of pollination to be carried out, $P_c$ is set in the range [0..1]. To start with, $P_c$ is set to 0.5 as initial value and then several experiments are carried out in the proposed hybrid metaheuristics to determine the best $P_c$ value. The best fitness value with minimal computation time is obtained when $P_c$ value is kept as 0.9 for all job sizes. Hence $P_c$ takes the value as 0.9.

3.4.5 Pollination Strategy

The ultimate objective of a flower is reproduction via pollination. Hence the proposed hybrid metaheuristics undergoes two types of pollination namely global pollination and local pollination. The global and local pollination is equivalent to crossover and mutation operation in GA respectively.

3.4.5.1 Global Pollination

Generate a float random number $rand$ in the range[0..1]. If the generated $rand$ value goes below switch probability ($P_c = 0.9$), then the pollen flowers undergo global pollination satisfying rule 1. The objective of doing global pollination for $P_c$ value < 0.9 is to generate diversified population. During global pollination process, the two job sequences selected are (1) best sequence $g_{best}$ which acts as parent 1 and (2) current pollen in that generation acts as parent 2. Both the selected parents undergo Partially Mapped Crossover (PMX) to generate a new offspring.
Partially Mapped Crossover (PMX) works by choosing a segment from parents at random and exchange subgroup between parents. The next step is to determine a mapping relationship with the elements in the random segment and an offspring is created using this relationship. After dealing with the elements from the crossover segment, the rest of the offspring can be filled from the second parent. An example for global pollination is given in Table 3.2.

**Table 3.2 Performing global pollination using Partially Mapped Crossover**

<table>
<thead>
<tr>
<th>Parent 1 $g_{best}$</th>
<th>4 2 1 3 10 7 9 6 5 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent 2 (Current Pollen):</td>
<td>4 2 3 7 6 9 5 8 1 10</td>
</tr>
<tr>
<td>Random segments : 1 to 5</td>
<td></td>
</tr>
<tr>
<td>Offspring :</td>
<td>4 2 1 3 10 9 5 8 7 6</td>
</tr>
</tbody>
</table>

Instead of narrowing down the search space, $n$ individuals work towards improving the fitness of GFPA whereas the remaining $n$ individuals randomly pollinate following traditional GA. Both optimization algorithms take equal probability to explore the search space.

A float random number $rand1$ is generated in the range $[0..1]$. If the generated $rand1$ value falls below 0.5, then the current individual undergoes global pollination with the randomly selected individual to generate a new offspring. If $rand1$ falls above 0.5, the current individual undergoes global pollination with the best sequence i.e $g_{best}$ to generate a new offspring. Each generation produces $2n$ individuals out of which $n$ individuals work together to improve the current $g_{best}$ while the remaining $n$ individuals explore the search space effectively.

Fitness evaluation is carried out for the newly generated offspring using fitness function. After each generation, job sequences with minimum
fitness values alone are retained. During global pollination process, the hybrid metaheuristics guarantees the exclusion of infeasible pollen sequences in the population. If two job sequences with same $g_{best}$ values are chosen for global pollination, then the hybrid metaheuristics undergoes local pollination irrespective of switch probability.

### 3.4.5.2 Local Pollination

If the generated $rand$ value is greater than 0.8, then the selected job sequence undergoes local pollination in which positions of the two jobs are swapped based on their random values and results in a new offspring by satisfying rule 2. Fitness function is evaluated for the newly generated offspring. If the fitness function of the newly generated offspring is a better one, then it is replaced with the current sequence or else it is discarded. An example for local pollination is given in Table 3.3.

<table>
<thead>
<tr>
<th>Parent 1 (Current Pollen): 4 2 3 7 6 9 5 8 1 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random segments : 1 and 5</td>
</tr>
<tr>
<td>Offspring : 6 2 3 7 4 9 5 8 1 10</td>
</tr>
</tbody>
</table>

During local pollination process, the hybrid metaheuristics guarantees the exclusion of infeasible pollen sequences in the population. Thus, GFPA is restricted to survival of fittest.
3.4.6 Termination of GFPA

Termination is important for any hybrid metaheuristics as this criterion determines how long an algorithm needs to run to generate quality solutions. The desirable properties of a termination criteria should address the following. The criterion should be (1) Easy to implement. (2) Able to provide stopping time automatically for any fitness function and (3) Should lead to good results. Hence an improvement-based automatic termination scheme has been proposed for the sequential hybrid metaheuristics to avoid unnecessary computations thereby resulting in reduction of computation time. Termination criterion is explained in detail in Chapter 4.

3.5 PSEUDOCODE OF GFPA FOR RSMCDD SCHEDULING

Pseudocode of the proposed algorithm to solve restricted single machine common due date scheduling problem is given in Algorithm 3.1. The convergence procedure for termination of algorithm is explained in Chapter 4.

Algorithm 3.1 Pseudocode of the Proposed Genetic Flower Pollination Algorithm

1: procedure GFPA(INT N, INT P_c)
2: input : n, p_c \quad \triangleright n: job size, P_c: Switch Probability
3: input : Set threshold = 0.5 \quad \triangleright Equal probability
4: output : Return best sequence g_{best}
5: for (i = 1 to 2n pollen in the population) do
6: Representation of 2^n pollen flowers (x_1, x_2, ..., x_{2n}) \quad \triangleright Initial population construction
7: Evaluate fitness function for pollen flowers  ▶ Fitness function evaluation
8: end for
9: while (!convergence()) do  ▶ Invokes automatic termination every time
10: Choose the current best sequence $g_{best}$ from the population
11: for (i = 1 to 2n pollen in the population) do
12: Generate a float random number rand in the range [0..1]
13: if (rand > switch probability) then  ▶ Local pollination
14: Perform local pollination (Swapping two jobs randomly)
15: Select two pollens randomly from the flowers and swap
16: Generate a new offspring
17: else
18: Perform global pollination (PMX Crossover)  ▶ Global pollination
19: Generate a float random number rand1 in the range [0..1]
20: if (rand1 > threshold) then
21: Parent 1 ← $g_{best}$
22: Parent 2 ← current pollen
23: if ($g_{best}$ = = current pollen) then
24: Perform local pollination
25: else
26: Generate a new offspring
27: end if
28: else
29: Parent 1 ← rand1
30: Parent 2 ← current pollen
31: if (Parent 1 = current pollen) then
32: Perform local pollination
33: else
34: Generate a new offspring
35: end if
36: end if
37: end if
38: if (fitness(new offspring) < fitness(current pollen)) then
39: current pollen ← offspring
40: end if
41: end for
42: end while
43: Display the best sequence found
44: Return best sequence $g_{best}$ ▷ Returns the best sequence found
45: end procedure
The input arguments for the procedure GFPA are job size $n$, switch probability $P_c$ and 280 benchmark problem instances. This procedure returns the best sequence as output. Initial population construction is explained in Section 3.4.2. Fitness function is evaluated for the pollen sequences. To explore the job sequences in all possible dimensions, a user-defined threshold is set not to limit the pollination strategy with respect to current $g_{best}$. This algorithm performs local pollination whenever current pollen and $g_{best}$ are same. Global pollination strategy is explained in Section 3.4.5.1. The algorithm terminates when the optimal solution is achieved.

3.6 SUMMARY

Genetic algorithms are not suitable for solving larger job size scheduling problems thereby paving way to the development of hybrid metaheuristics to provide high quality solutions in an acceptable time. The proposed work has led to the development of a novel nature-inspired evolutionary hybrid metaheuristics namely Genetic flower pollination algorithm to tackle 280 benchmark problems of RSMCDD scheduling. The research work has also attempted to suggest an automatic termination algorithm for the hybrid metaheuristics for faster execution of large job sizes and is discussed in detail in the next chapter.