**Appendix – Pseudo Codes**

**Listing 1.** Pseudocode of the proposed ABCA based algorithms

**Input:** Iteration number , number of employed bees , the number of onlooker bees , scout bee limit

**Begin**

; // Iteration counter

; // Counter for non-improving solutions

The population is initialized using the specified initial population generation scheme;

The fitness function (FF) values for the initial population are computed;

**While** **do**

**For** // Employed bee phase

A food source is selected randomly;

A neighboring solution is generated by applying the chosen *crossover operator*;

The FF value of the new solution is computed;

The *mutation operator* is applied to improve the solution;

**If** the solution is improved **then**

The employed bee is updated;

**Else**

;

**End If**

**End For**

**For** // Onlooker bee phase

A solution using the *roulette wheel selection* operator;

A new solution is generated by applying the *crossover operator*;

The FF value of the new solution is computed;

**If** the solution is improved **then**

The onlooker bee is updated;

**Else**

;

**End If**

**End For**

**For** // (Scout bee phase)

**If**  **then**

The current solution is abandoned;

A new solution is generated using the *initial population generation* scheme;

**Else**

The existing solution is preserved;

**End If**

**End For**

;

**End While**

The best solutions found at each iteration are returned;

**End**

**Output:** The best solution and corresponding fitness values

**Listing 2.** Pseudocode of the proposed decoding procedure

**Input:** the family set of all sublots, the set of all routes, the number of sublots from a lot **,** thetotalnumber of sublots , the total number of workers , the capacity of vehicles , the sublot sizes for each scenario , the sublot size for robut part (), the set of scenarios and associated probabilities , the level of robustness

**Begin**

**For**  // For the stochastic component

;

**While** **do**

Sublot is removed from according to the chromosome order;

The processing time of the sublot is computed;

Possible starting and completion times for sublot are determined;

A list of possible starting and completion times is formed for sublot ;

**While** **do**

Anypreviously assigned sublots overlapping with sublot , based on , are identified;

The total number of workers required for the earliest starting time in is computed;

**If** is not exceeded **then**

The earliest feasible starting time is assigned to sublot ;

The completion time of sublot is computed;

**Break While;**

**Else**

The earliest starting time and relevant completion time option are removed from ;

**End If**

**End While**

A list of starting and completion times of scheduled sublots is formed;

**If** the RVA approach is employed for distribution scheduling **then**

Sublot is assigned to the first vehicle in ;

The relevant vehicle’s capacity is updated;

**Else If** FFFAVA approach is employed for distribution scheduling **then**

The sublot with the earliest completion time in is assigned to the first vehicle in ;

The relevant vehicle’s capacity is updated;

**End If**

**If** vehicle’s capacity is exceeded **then**

The relevant vehicle is removed from ;

**End If**

;

**End** **While**

The starting and completion times of all sublots in production scheduling are returned;

The departure time of the relevant vehicle is set to the latest completion time;

The starting and completion times for distribution scheduling of all sublots are computed;

The expected maximum lead time for the stochastic part is computed;

**End For**

The same procedure that is applied for each scenario is also applied to the robust component;

The maximum lead time for the robust part is computed;

The fitness function is computed by considering ;

**End**

**Output:** The fitness function value

**Listing 3.** Pseudocode of the initial population generation operator

**input:** , ,,,the lot sizes , , ,

**Begin**

**For**

Each lot is divided into sublots;

**If** RSS is applied for sublot sequencing **then**

Sublots are ordered randomly, and the ordered set of all sublots is obtained;

**Else** // HSS is applied for sublot sequencing

Sublots are ordered based on Heijunka, and the ordered set of all sublots is obtained;

**End If**

The first section is generated based on ;

Sublots are assigned to the serus randomly for the second section;

**For** d // For the stochastic component

Seru modes are generated randomly for each sublot-seru assignment;

Sublot sizes of each lot are generated randomly;

Vehicle orders are generated randomly;

**End For**

Seru modes, sublot sizes, and vehicle orders for the robust component are generated;

The chromosome is constructed by combining all sections;

**End For**

**End**

**Output:** The initial population for the chosen approach

**Listing 4.** Pseudocode of thecrossover operator

**Input:** The set of alternative cut points , parents

**Begin**

The *assignment of lots/sublots* section of offspring is taken from parent ;

The *assignment of lots/sublots* section of offspring is taken from parent ;

The *assignment of seru section* of offspring is taken from parent ;

The *assignment of seru section* of offspring is taken from parent ;

**For** d // For the stochastic component

The *modes of seru* section of offspring is taken from parent ;

The *modes of seru* section of offspring is taken from parent ;

The *assignment of vehicle* section of offspring is taken from parent ;

The *assignment of vehicle* section of offspring is taken from parent ;

**If** one-point crossover operator is applied **then**

A random number is chosen between 1 and ;

**For**

The sublot sizes of lot for the offspring are taken from parent ;

The sublot orders in offspring are arranged to remain consistent with parent ;

**End**

**If**  **then**

**For**

The sublot sizes of lot for the offspring are taken from parent ;

The sublot orders in offspring are arranged to remain consistent with parent ;

**End For**

**End If**

The same procedure is applied for the offspring **;**

**End For**

**Else //**two-point crossover operator is applied

Two random numbers and are chosen between 1 and ;

**For**

The sublot sizes of lot for the offspring are taken from parent ;

The sublot orders in offspring are arranged to remain consistent with parent ;

**End**

**For**

The sublot sizes of lot for the offspring are taken from parent ;

The sublot orders in offspring are arranged to remain consistent with parent ;

**End**

**For**

The sublot sizes of lot for the offspring are taken from parent ;

The sublot orders in offspring are arranged to remain consistent with parent ;

**End**

**End If**

**End For**

Apply the same procedure applied for each scenario for the robust component;

**End**

**Output:** Offsprings and