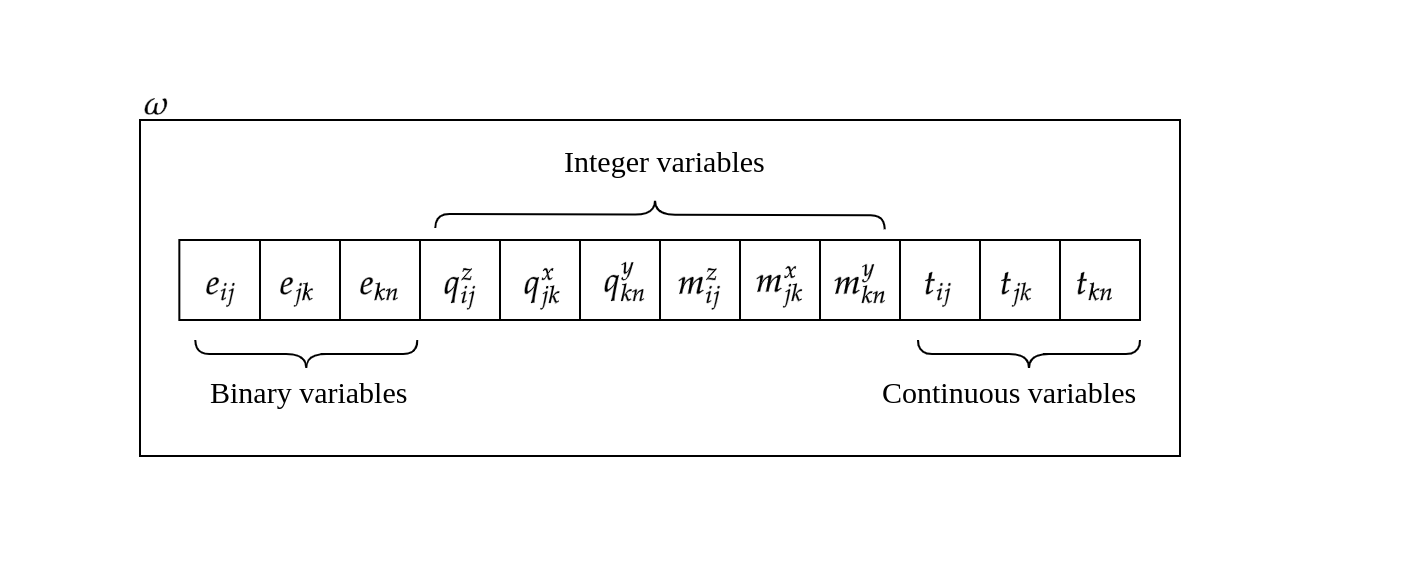
## CRO Solution Representation and Analysis

#### Encoding Process

During the encoding of a molecule from the graph, we store all information as a vector of the required decision variables.

The vector format, represented by , is a 1D representation of all the graph variables. It can be produced by flattening each variable in the intermediate form and then concatenating each of the flattened variables to each other in order to get a 1D vector containing all necessary information about the supply chain network graph. This encoded vector acts as the molecular structure of each molecule in the CRO algorithm.



We also store an set of intermediate matrices, the route activity (total number of ventilators being transported over each route), denoted by . All operators in the priority based CRO algorithm operate upon the route activity, after which we convert the transformed route activity into the molecular structure .

#### Interconversion process for and

In order to convert a route activity matrix into the required decision variables, we define a process called stratification. It is based on the priority constraints of vehicles.

Consider the connection between and . represents the total demand for ventilators across the route to . In order to calculate how many ventilators are carried by each type of vehicle, we use the previously mentioned priority constraints. That is, large vehicles are loaded first, until the remaining demand falls below that of the capacity of a single medium vehicle. Then, medium vehicles are loaded until the remaining demand falls below that of the capacity of a single small vehicle and the remaining capacity if any, is loaded onto small vehicles.

Thus, we can extract values for from . From the other two decision variables can be calculated. can be calculated as follows.

In order to perform the reverse process, that is to go from the decision variables to the route activity, we define a process called collapsing. We set values for the elements of as follows.

We take the sum of quantity of ventilators along the third dimension ( vehicle type dimension ) and convert the 3 dimensional decision variable to our 2 dimensional route activity.

#### Parameters Used

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| *buffer* | 2300 |
| *InitialKE* | 400 |
| *MoleColl* | 0.7 |
| *KELossRate* | 0.25 |
|  | 400 |
|  | 40 |

#### CRO Algorithm

#### Initialization

We start by defining our objective function, constraints, and the dimensions of our solution vector space to the CRO algorithm. In this stage, we would also have to assign values to the variables and hyperparameters ( *KELossRate, MoleColl, buffer, InitialKE* , and ).

Each molecule possesses a structure ( ) which is a vector of the dimensions of our required solution. We first generate a molecule in intermediate format ( ).

Considering the connection between *I* and *J*, we first initialize the binary variable by generating a random real value between 0 and 1 and rounding to the nearest integer. If is 1, it implies that the current solution has an active route between and . When initializing , we assign 0 to inactive routes (). For the remaining active paths, we assign a random value to each of them such that their sum is constant, so as to satisfy constraint 8,9,10. Next, we initialize and randomly such that the constraints concerning them are also satisfied.

#### Iteration

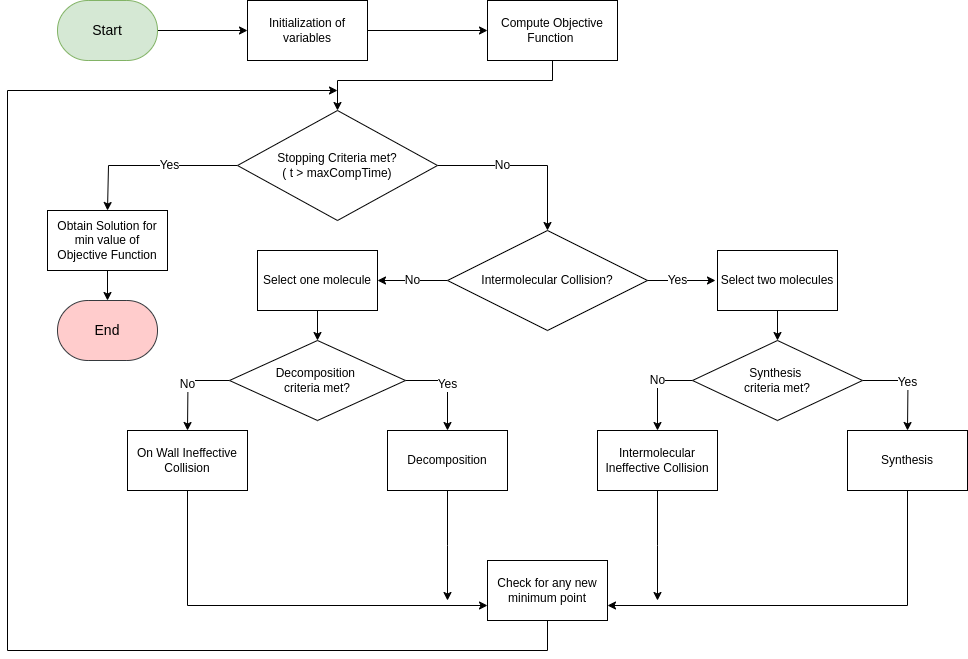
As per the CRO algorithm, at each iteration each molecule undergoes one of the 4 possible steps : *OnWallIneffectiveCollision*, *IntermolecularIneffectiveCollision*, *Decomposition* and *Synthesis*. In order to decide whether the reaction would be unimolecular or intermolecular, a random number is generated. If > MoleColl or if the system has one molecule, a unimolecular reaction proceeds. Else, a intermolecular reaction proceeds.

Within the unimolecular reactions, the criterion for a decomposition reaction is *NumHit* - *MinHit* > . Else, an on-wall ineffective collision follows.

For the intermolecular reactions, a synthesis takes place when the KE of reacting molecules are less than . Otherwise, an ineffective intermolecular collision follows.

#### Termination

The algorithm terminates once the time exceeds the maximum computing time assigned. The best solution found and its corresponding objective function value among all molecules is then outputted.



#### Neighbourhood Search Operators

#### 2-opt

Consider the connection from *I* to *J*. A 2-opt swap exchanges the activity of two routes between *I* and *J*. That is, for and randomly selected,

Now, this process is repeated for connections between *J* and *K*, and *K* and *N* to complete the exchanges across all connections and get the new intermediate output. It is denoted as *2-opt*().

#### Gaussian Mutation (with reflection)

This operator is used to vary the integer and continuous variables. Consider the connection from *I* to *J*. We generate a random array with order same as with elements drawn from a normal distribution . We add this change to our variable, rounding to the nearest integer on an integer variable. In order to satisfy the non negativity constraint, we take the absolute value of each element.

We repeat this process for connections between *J* and *K*, and *K* and *N* to get the new intermediate output after collision. It is denoted as GMR().

#### Decomposition Operator

During decomposition, the two new molecules are created from the original molecule. In order to produce a new solution, say , we would first copy the route activity of original molecule into . That is,

One of the three layers in is then selected to be retained, while the other two layers are randomized. Thus, the new molecule produces a new solution while simultaneously retaining certain information of the parent molecule. We apply the same process to generate . The two newly generated route activities are returned by the operator

This operator requires the molecular structure in intermediate format as input. It is denoted as *D*().

#### Probabilistic Select

During synthesis, we combine two molecular structures, say and , to generate a new molecular structure, say . Consider the connection from *I* to *J*. We randomly generate a binary matrix *c* of order same as . Thus, . If is 0, then the value for is taken from , and otherwise it is taken from .

This operator requires the molecular structure in vector format as input. It is denoted as *PS*(, ).

#### Pseudocodes

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm 1: OnWallIneffectiveCollision** | | | |
| **1** | **Input** : Molecule | | |
| **2** | Collapse to route activity matrix | | |
| **3** |  | | |
| **4** | Stratify to molecular structure | | |
| **5** |  | | |
| **6** |  | | |
| **7** | **if** **then** | | |
| **8** |  | Randomly generate | |
| **9** |  |  | |
| **10** |  |  | |
| **11** |  |  | |
| **12** |  |  | |
| **13** |  |  | |
| **14** |  | **if** **then** | |
| **15** |  |  |  |
| **16** |  |  |  |
| **17** |  |  |  |
| **18** |  | **end if** | |
| **19** | **end if** | | |

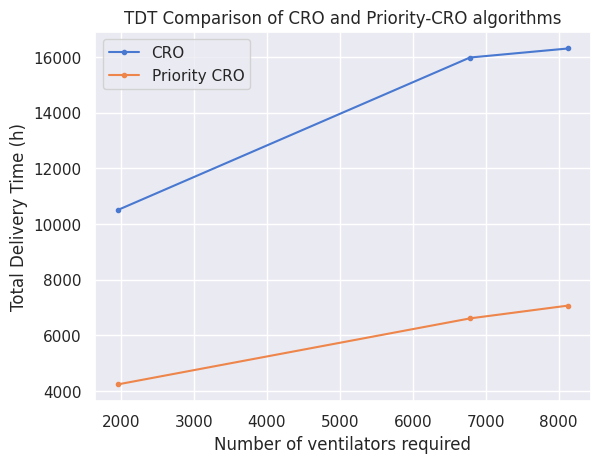
|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm 2: IntermolecularIneffectiveCollision** | | | |
| **1** | **Input** : Molecules and | | |
| **2** | Collapse to route activity matrices respectively. | | |
| **3** |  | | |
| **4** |  | | |
| **5** | Stratify , to molecular structures | | |
| **6** | and | | |
| **7** | and | | |
| **8** |  | | |
| **9** | **if** **then** | | |
| **10** |  | Randomly generate | |
| **11** |  | and | |
| **12** |  | and | |
| **13** |  |  | |
| **14** |  |  | |
| **14** |  | **if** **then** | |
| **15** |  |  |  |
| **16** |  |  |  |
| **17** |  |  |  |
| **18** |  | **end if** | |
| **19** |  | **if** **then** | |
| **20** |  |  |  |
| **21** |  |  |  |
| **22** |  |  |  |
| **23** |  | **end if** | |
| **24** | **end if** | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm 3: Decomposition** | | | |
| **1** | Input : Molecule | | |
| **2** | Create empty Molecules | | |
| **3** | Collapse to route activity matrix | | |
| **4** | , | | |
| **5** | Stratify , to molecular structures | | |
| **6** | and | | |
| **7** | **if** **then** | | |
| **8** |  |  | |
| **9** |  | **goto** step 19 | |
| **10** | **else** | | |
| **11** |  | Randomly generate | |
| **12** |  |  | |
| **13** |  | **if** **then** | |
| **14** |  |  | Delete Molecules |
| **15** |  |  |  |
| **16** |  | **else** | |
| **17** |  |  |  |
| **18** |  |  | Randomly generate |
| **19** |  |  |  |
| **20** |  |  |  |
| **21** |  |  | and |
| **22** |  |  | and |
| **23** |  |  | and |
| **24** |  |  | Delete |
| **25** |  | **end if** | |
| **26** | **end if** | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm 4: Synthesis** | | | |
| **1** | Input : Molecules and | | |
| **2** | Create empty Molecule | | |
| **3** | Collapse to route activity matrices respectively. | | |
| **4** |  | | |
| **5** | Stratify to molecular structure | | |
| **6** |  | | |
| **7** |  | | |
| **8** | **if** **then** | | |
| **9** |  |  | |
| **10** |  |  | |
| **11** |  |  | |
| **12** |  |  | |
| **13** | **else** | | |
| **14** |  | and | |
| **15** |  | Delete | |
| **16** | **end if** | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Algorithm 5: CRO Algorithm** | | | | |
| **Require:** *maxTime, popSize* | | | | |
| **1** | Set initial values of | | | |
| **2** | Create *popSize* number of molecules | | | |
| **3** | **while** **do** | | | |
| **4** |  | Sample | | |
| **5** |  | **if**  **then** | | |
| **6** |  |  | Randomly select one molecule | |
| **7** |  |  | **if** **then** | |
| **8** |  |  |  | Proceed with *Decomposition* |
| **9** |  |  | **else** | |
| **10** |  |  |  | Proceed with *OnWallIneffectiveCollision* |
| **11** |  |  | **end if** | |
| **12** |  | **else** | | |
| **13** |  |  | Randomly select two molecules and | |
| **14** |  |  | **if**  **then** | |
| **15** |  |  |  | Proceed with *Synthesis* |
| **16** |  |  | **else** | |
| **17** |  |  |  | Proceed with *IntermolecularIneffectiveCollision* |
| **18** |  |  | **end if** | |
| **19** |  | **end if** | | |
| **20** | **end while** | | | |
| **21** | Return best solution and minimum *TDT* | | | |

#### Additional Graphs and Tables



|  |  |  |  |
| --- | --- | --- | --- |
|  | Priority CRO | CRO | Speedup Factor |
| Wave 1 | 4235.21 | 10497.93 | 2.48 |
| Wave 2 | 6607.85 | 15979.93 | 2.42 |
| Wave 3 | 7069.74 | 16305.16 | 2.31 |