**Table I.** Scheduling

|  |
| --- |
| **Input:** //’s are the low capacity trucks moving in local roads |
| //’s are the high capacity trucks moving in highways |
| **Output:** //Shortest path route from to matching point |
| //Shortest path route from to matching point |
| **While** **do** |
| **If** **then** //Capacity of reaches a threshold |
| //get emptied in the dump |
| **End If** |
| **If** **then** //Capacity of reaches threshold |
| //Find the nearest to |
| //Find the matching point in the highway |
| //Compute the shortest path route from to |
| //Compute the shortest path route from to |
|  |
|  |
| **End If** |
| **End While** |

**Table II.** Nearest

|  |
| --- |
| **Input:** //Low capacity truck moving in local roads |
| //’s are the high capacity trucks moving in highways |
| **Output:** //Nearest to |
| //Set the first |
| //Minimum distance between and |
| **For** **do** //Find the with minimum distance from |
| //Get the next |
| **If** **then** |
|  |
| //Get the nearest to |
| **End If** |
| **End For** |

**Table III.** Match

|  |
| --- |
| **Input:** //Low capacity truck moving in local roads |
| //High capacity truck moving in highways nearest to |
| **Output:** //Crossroad match point in junction of local road and highway between and |
| // Crossroad points in junction of local roads and highways between and |
| //Set the first |
| //Minimum distance between and |
| //Minimum distance between and |
| **For** **do** //Find the match point with minimum distance between both and |
| //Get the next |
| **If** **then** |
|  |
|  |
| //Get the nearest from both and |
| **End If** |
| **End For** |

**Table IV.** Routing

|  |
| --- |
| **Input:** // is either a or truck |
| **Output:** //route from to . Implements the shortest path routing algorithm in [1] |
| //Initialize the set of vertices to be equal to the set of crossroads between and |
| //Initialize the set of links to be equal to the set of roads between and |
| //Initialize the set of weights to be equal to the set of distances between and |
| //Compute route according to the customized algorithm in [1] |

**Experiments**

1. Measure the frequency in which the scheduling is triggered
2. Measure the average time between scheduling triggers
3. Measure response time between a trigger and its handling from
4. Measure response time between a trigger and its handling from
5. Measure the average number of ’s served by a certain number of
6. Measure capacity, distance, time, fuel required during
7. Measure capacity, distance, time, fuel required during
8. Measure CPU elapse time of scheduling algorithm

**Simulation Numbers**

1. ***HCT Scheduling.***
   1. *Simulation values for HCTs:*
      1. capacity -> 11000 to 12000
      2. distance -> 10 to 15
      3. time -> 20 to 30
      4. fuel -> 2 to 4
2. ***LCT Scheduling.***
   1. *Simulation values for HCTs:*
      1. capacity -> 1 to 12000
      2. distance -> 5 to 10
      3. time -> 10 to 15
      4. fuel -> 1 to 3
   2. *Simulation values for LCTs:*
      1. capacity -> 1 to 3000
      2. distance -> 2 to 7
      3. time -> 7 to 12
      4. fuel -> 1 to 2

**References**

1. P. Avella, M. Boccia and A. Sforza, “Resource Constraint Shortest Path Problems in Path Planning for Fleet Management”, Journal of Mathematical Modeling and Algorithms, Kluwer Academic Publishers, 2004, vol. 3, pp. 1-17.