# Appendix A. Tables

## Table A.1. Computation expressions

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Name** | | **Expression** | | | **（intermediate variable）** | | |
| ) | |  | | | - | | |
| 3) | |  | | |  | | |
| 4) | |  | | | - | | |
| 5) | |  | | |  | | |
| 6) | |  | | |  | | |
|  | |  | | |  | | |
| 8) | |  | | | - | | |
| 9) | |  | | |  | | |
| 10) | |  | | |  | | |
|  | |  | | |  | | |
|  | |  | | | - | | |
|  | |  | | |  | | |
|  | |  | | |  | | |
|  | |  | | |  | | |
|  | |  | | |  | | |
|  | |  | | | - | | |
|  | |  | | |  | | |
|  | |  | | |  | | |
|  | |  | | | - | | |
|  | |  | | |  | | |
|  | |  | | |  | | |
|  | |  | | |  | | |
|  | |  | | |  | | |
|  | |  | | | - | | |
|  | |  | | |  | | |
|  | |  | | |  | | |
|  | |  | | |  | | |
|  | |  | | |  | | |
|  | |  | | |  | | |
|  | |  | | |  | | |
|  | |  | | |  | | |
|  | |  | | |  | | |
|  | |  | | |  | | |
|  | |  | | | - | | |
|  | |  | | |  | | |
|  | |  | | |  | | |
|  | |  | | |  | | |
|  | |  | | | - | | |
|  | |  | |  | | |
|  | |  | | - | | |
|  | |  | |  | | |
|  | |  | |  | | |
|  | |  | |  | | |
|  | |  | |  | | |
|  | |  | |  | | |
|  | |  | |  | | |
|  | |  | | - | | |
|  | |  | |  | | |
|  | |  | | - | | |
| 1) | |  | | - | | |
|  | |  | | - | | |
|  | |  | | - | | |
|  | |  | | - | | |
|  | |  | | - | | |
|  | |  | | - | | |
|  | |  | | - | | |
|  | |  | | - | | |
|  | |  | | - | | |
|  | |  | | - | | |
|  | |  | | - | | |
|  | |  | | - | | |
|  | |  | | - | | |
|  | |  | |  | | |
|  | |  | | - | | |
|  | |  | |  | | |
|  | |  | |  | | |
|  | |  | |  | | |
|  | |  | |  | | |
|  | |  | |  | | |
|  | |  | |  | | |
|  | |  | | - | | |
|  | |  | |  | | |
|  | |  | |  | | |
|  | |  | |  | | |
|  | |  | | - | | |
|  |  | |  | | |
|  |  | |  | | |
|  |  | | - | | |
|  |  | |  | | |
|  |  | | - | | |
|  |  | |  | | |
|  |  | |  | | |
|  |  | | - | | |
|  |  | | - | | |

## Table A.2. Conflict patterns between movements



## Table A.3. Computation expressions of radius in an unsignalized intersection

|  |  |
| --- | --- |
| **Name** | **Expression** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Appendix B. Algorithms

## Algorithm B.1. The crossing strategy of an AV

|  |  |
| --- | --- |
|  | // is the amount of preceding vehicles.  // is the initial time.  // is the fastest entry time of the vehicle arriving at the intersection without delay.  // , , are the initial speed, entry speed, and limited speed, respectively |
| 1  2  3  4  5  6  7  8  9  10  11 | **Begin**  Calculate the fastest entry time of the new incoming CAV by Eq. (9);  Initialize the permissive entry time ;  **For** *k =* 1 to **Do**  **If** its spatial trajectory is in conﬂict with that of the preceding vehicle *k* **Then**  **If**  *< <* , where **Then**  ;  **End if**  **End if**  **End for**  **End** |

## Algorithm B.2. Game process of multiple firms for trajectory planning

|  |  |
| --- | --- |
|  | // is a small positive constant.  // is the simulated traffic state at time .  //Before time , is the set of travel requests of firm during .  //At time , trajectories for are planned. |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45 | **Begin**  Firm creates a sequence set for based on the departure time sequence;  Firm creates an empty sequence set to place the final AV sequence for ;  ;  **Repeat**  **If** **Then**  **For** firm **Do**  Copy a simulated traffic state based on ;  **For** AV **Do**  Plan the trajectory ;  Update by the trajectory;  **End for**  **End for**  **Else**  **For firm Do**  Sort delays of AVs in in descending order;  　　　 Addthe AVs with the largest delays into  and remove them from  Copy a simulated traffic state based on ;  **For** AV **Do**  Plan the trajectory ;  Update by the trajectory;  **End for**  **For** AV **Do**  Plan the trajectory ;  Update by the trajectory;  **End for**  **End for**  **End if**  The SMC collects all requests and creates a checking sequence set ;  Create an empty set to place the AVs who have been checked for safety;  **For** AV **Do**  Modify by Appendix B.1 if there is a conflict between and any AV;  Add into ;  **End for**  The SMC feeds back feasible trajectories and the average system delay ;  Create an empty firm set ;  **For** firm **Do**  　　 Calculate the average delay ;  **If** and **Then**  　　　 Add into ;  **End if**  **End for**    **Until**  **End** |

## Algorithm B.3. The fair strategy

|  |  |
| --- | --- |
|  | // is the iteration count.  **//**Given the AV sequence set and the number of travel requests of firm ; |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | **Begin**  Create an empty sequence set to place the checking sequence for all AVs;  Find the greatest common divisor of ;  Round up the quotient to the integer , ;  Find firm who has the most requests;  **For** **Do**  Create an empty sequence set to place the checking sequence for AVs;  Add the first AVs in into in turn and remove them from ;  **For** firm **Do**  **If** **Then**  Randomly generate an index set consisting of non-repetitive elements in  ;  Rearrange the numbers in in ascending order;  Add the first AVs in into according to and remove these AVs from ;  **End if**  **End for**  Add the AVs in into in turn;  **End for**  **End** |

## Algorithm B.4. The partial priority strategy

|  |  |
| --- | --- |
|  | // is the iteration count.  **//**Given the AV sequence set and the number of travel requests of firm ; |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | **Begin**  Create an empty sequence set to place the checking sequence for all AVs;  Find the greatest common divisor of ;  Round up the quotient to the integer , ;  Find firm who has the most requests;  **For** **Do**  Create an empty sequence set to place the checking sequence for AVs;  Add the first AVs in into in turn and remove them from ;  **For** firm **Do**  **If** **Then**  Randomly generate an index set consisting of non-repetitive elements in  ;  Rearrange the numbers in in ascending order;  Add the first AVs in into according to and remove these AVs from ;  **End if**  **End for**  Add the AVs in into in turn;  **End for**  **End** |