单节点局部最优化之枚举策略(Demo\_v1.java)：

因为是单节点局部最优化，即每个十字路口（或T字路口）之间是独立的（不考虑当前节点流过的车流量对流向的路口下一时间点流量的影响），那么只需要求出当前时刻使得每一个十字路口的车辆等待时间最小。又因为右转不违反交通规则，所以右转总是可行，那么每个路口只有左转与直行的状态不确定，因此每个十字路口的状态个数为4\*4\*4\*4=256种，T字路口为4\*4\*4=64种。因为状态数不是很多，所以可以使用枚举，即对每个节点，枚举每一种状态，选择最优的一个状态作为该节点的状态。

单节点局部最优化之随机搜索策略(Demo\_v2.java)：

因为是单节点局部最优化，即每个十字路口（或T字路口）之间是独立的（不考虑当前节点流过的车流量对流向的路口下一时间点流量的影响），那么只需要求出当前时刻使得每一个十字路口的车辆等待时间最小。又因为右转不违反交通规则，所以右转总是可行，那么每个路口只有左转与直行的状态不确定，因此每个十字路口的状态个数为4\*4\*4\*4=256种，T字路口为4\*4\*4=64种。如果采用枚举算法，那么每个十字路口（T字路口）需要枚举256次（64次），为了稍微降低时间复杂度以及局部最优并不能保证全局最优，所以这里采用随机搜索策略，即对于每个十字路口或者T字路口随机SEARCH\_NUM次(取200次)，选取最好的一次作为该节点的状态。

通过运行两个版本得到的结果，随机搜索策略效果要比枚举策略好，因为每个时间点局部最优并不能保证全局最优，如果想要保证全局最优，那么需要根据历史流量对将来的流量进行预测。