Sabancı University IE Optimization Challenge - Final Round Problem

FreshCart's success to deliver goods in a timely manner has skyrocketed the orders, putting more pressure on the company's operations team and the delivery personnel. The company's legal team is in the process of finalizing a contract with a third party logistics provider to effectively scale the company's operations while maintaining service quality and alleviating the strain on its internal resources. However, this contract will not be in effect for at least another few weeks. As a member of the operations team, your top priority is to keep high levels of customer satisfaction, but you also would like to ensure that the delivery personnel is not overwhelmed by the rapidly increasing workload.

To use the available resources efficiently, you came up with an approach which entails somewhat relaxing the time window restrictions when planning the delivery schedules. More specifically, you allow early/late deliveries at a certain cost if it facilitates a better overall solution. The cost associated with making an early delivery (i.e. before the time window opening of a customer) is c_e per unit time. Say there is a customer whose time window opens at time 120, but the delivery vehicle will be ready at the customer location at time 100. If the delivery takes place at time 100, then the cost incurred by this early delivery will be $(120 - 100) * c_e$. Similarly, the cost associated with making a late delivery (i.e. after the time window closing of a customer) is c_l per unit time. Note that c_l is much higher in value than c_e so that the late deliveries are discouraged more and only used as a last resort. Despite the flexibility achieved by allowing early/late deliveries, you know that it is important not to deviate too much from the originally given customer time windows for service quality reasons. Hence, you restrict the amount of earliness/lateness allowed for each customer to 20% of the length of his/her time window. In particular, the earliest and the latest times that a customer with time window $[e_i, l_i]$ can be served are $e_i - \lceil 0.2(l_i - e_i) \rceil$ and $l_i + \lceil 0.2(l_i - e_i) \rceil$, respectively. Note here that for any customer, the service is assumed to take place starting at the earliest time possible upon the arrival of the assigned vehicle, and as soon as it is completed, the vehicle must depart from the customer location and travel to the next customer in its route. The vehicles are allowed to return to the depot (which is represented by node 0 in the instance files) later than its time window closing, again at a unit cost of c_l, and there is no limit on the amount of lateness for the depot. When solving the problem please take $c_e = 0.2$ and $c_l = 1$.

What you need to submit:

1. Your code file(s)
2. Your solution files in a zip folder (see the instance folder for a sample solution file, you must have a solution file in the same format for every instance you solve) - make sure to submit a separate solution file for each individual instance, and name it using the convention **sol_instance_id** where id is the id number of the instance (ranging from 1 to 20). For example, your solution file for instance_8 should be named as sol_instance_8.
3. A slide deck you will use to present your solution approach & results

Evaluations will be based on the quality of the solutions submitted. The quality of a solution is measured by its cost; solutions having lower cost values are considered to be of higher quality. The cost of a solution is now defined by three components: (1) the total distance/travel time of the routes, (2) the fixed cost per each vehicle used, and (3) the total cost incurred by early/late deliveries. For different solutions of the same cost, the one having the lowest violation of the original time windows (i.e. total amount of earliness and lateness) will be favored over the others. In case of ties in both respects, we will consider the solutions involving the fewest number of routes to be the best ones. Similar to the first round of the challenge, you need to calculate & use the Euclidean distances (you may use one decimal point and truncation) between all pairs of locations for each problem instance, and take them as your distance/travel time values. For a problem instance with n customers, set the fixed cost of using a vehicle to be $2 * n * c_{max}$ where c_{max} is the maximum of the distance values connecting any pair of locations.