A Trilevel *r*-Interdiction Selective Multi-Depot Vehicle Routing Problem

Supplementary Material (B)

Iterative Marginal Cost Analysis

1-Node iMCA:

The marginal cost of a given customer i is defined as in Fig. A1 where k and l are the predecessor and successor of customer i, respectively. If i is not worth visiting, then it can be dropped from the route. In other words, customer i should be outsourced. Otherwise, the cost of visiting customer i turns out to be less than its outsourcing cost. Therefore, visiting customer i is desirable, and it should stay between customers k and l. The pseudo code of 1-Node iMCA is provided in Table A1. The pseudo code of 2-Node iMCA can be derived in a straightforward way from there.

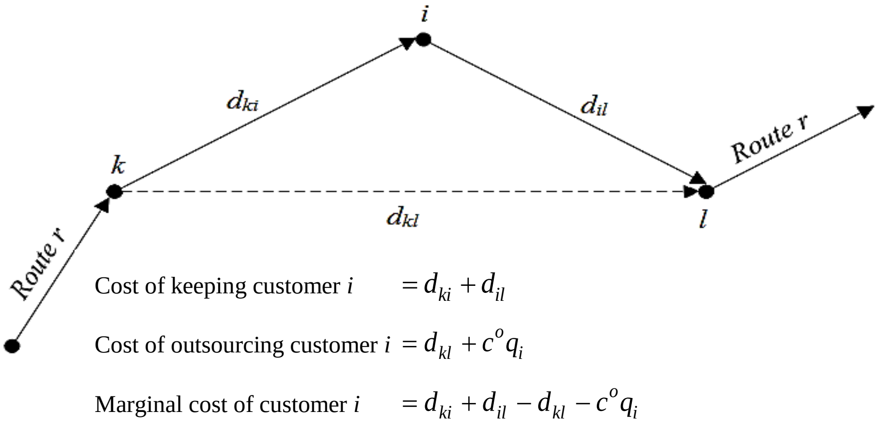
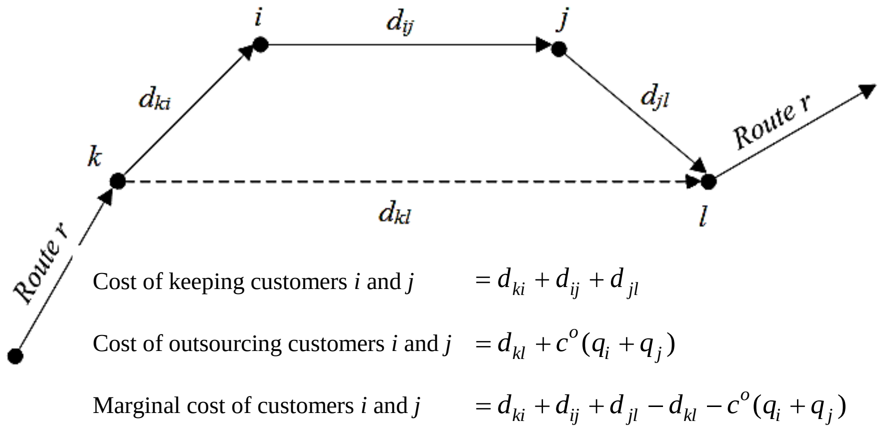

i
d_{ki}
d_{il}
k
l
d_{kl}
Route r
Route r
Cost of keeping customer i $= d_{ki} + d_{il}$
Cost of outsourcing customer i $= d_{kl} + c^o q_i$
Marginal cost of customer i $= d_{ki} + d_{il} - d_{kl} - c^o q_i$

Fig. A1. 1-Node Marginal Cost Analysis

2-Node iMCA:

The marginal cost of a given chain of two customers i and j is defined as in Fig. A2 where *k* and *l* are the predecessor and successor of customer i and customer j, respectively. If the chain of customers i and j is not worth visiting, then they can be dropped from the route. In other words, i and j should be outsourced. Otherwise, the cost of visiting this chain turns out to be less than its outsourcing cost. Therefore, visiting customers i and j is desirable, and they should stay as a chain between customers k and l .



Cost of keeping customers *i* and *j* $= d_{ki} + d_{ij} + d_{jl}$

Cost of outsourcing customers *i* and *j* $= d_{kl} + c^{o}(q_i + q_j)$

Marginal cost of customers *i* and *j* $= d_{ki} + d_{ij} + d_{jl} - d_{kl} - c^{o}(q_i + q_j)$

Fig. A2. 2-Node Marginal Cost Analysis

Table A1. The pseudo code of 1-Node iMCA

Notation

R : Current set of routes.

N_r : Subset of customers on route $r \in R$.

MC_i : Marginal cost of customer i .

$Index[MC_{[1]}]$: Index of customer with the highest marginal cost $MC_{[1]}$.

$succ(i),\ pred(i)$: Successor and predecessor of customer i , respectively.

```
**For** every route $r \in R$
    **For** every customer $i \in N_r$ on route $r$
        Set $MC_i = d_{ki} + d_{il} - d_{kl} - c^o q_i$ ;    // Compute marginal cost of each customer on route $r \in R$.
    **End For**
    Sort $MC_i$ values in nondecreasing order and create a sorted stack $S$ ;
    **While** ( $|N_r| > 0$ )
        Retrieve $MC_{[1]} = Pop(S)$ ;               // Return and remove the highest marginal cost.
        **If** $MC_{[1]} < 0$                          // Marginal cost of all customers are negative.
            **Break While** loop;                   // Stop the marginal cost analysis on the current route $r$ .
        **Else**
            Set $i_{[1]} = Index[MC_{[1]}]$ ;            // Customer $i$ with highest marginal cost.
            Remove $i_{[1]}$ from the route $r \in R$ ; // Remove the customer with the most positive MC.
            Update the $MC$ values of $succ(i_{[1]})$ and $pred(i_{[1]})$ ;
            Restore the nondecreasing order of $MC_i$ in the sorted stack $S$ ;
            Update $r \in R$ ;
            Update $N_r$ ;
            **If** $|N_r| = 0$                           // Route $r$ does not visit any customers.
                Discard route $r \in R$ ;
            **End If**
        **End If**
    **End While**
**End For**
```