A Multi-Start Iterative Local Search for the k-Traveling Repairman Problem

Francisco González, Juan Carlos Rivera
Functional Analysis and Applications Research Group
Universidad EAFIT
Medellín, Colombia

May 2015

1 Introduction

The combinatorial optimization problems as the distribution of goods, have a lot of applications in the industry, commerce and other activities. The importance of studying the distribution of goods is considerable. This paper is dedicated to the k-Traveling Repairman Problem (k-TRP), a vehicle routing problem in which a fleet of vehicles must visit a set of demand nodes. This problem is part of the current state of the art in areas like the operations research, applied mathematics and combinatorial optimization, and it has applications in several contexts as distribution with service-based factors, humanitarian logistics and green logistics. The ideal scenario is to handle this type of problems using integer linear programming models. However, because of their complexity and the size of the real problems, this kind of problems are usually solved by heuristic and metaheuristic methods.

In this paper, the design, experimentation, and analysis of a metaheuristic algorithm to solve the k-TRP is performed. The k-TRP is a variant of the traveling salesman problem (TSP) in which the cost-based classical objective function becomes the weighted sum of completion times at sites and k agents or vehicles cover one of k routes. The k-TRP is an optimization problem where k dealers, needs to schedule their visitation order to a set of customers, taking into account that each customer has a level of priority. In our implementation those priorities correspond to their demands . A solution is given by a schedule of the visits for each agent k, trying to find the set of least costly routes.

Taking into account that the TRP is an NP-Hard problem (Sahni & Gonzalez, 1974), and the k-TRP version studied in this paper generalizes TRP by including several agents and weights for each node, it can be concluded that our problem is also NP-Hard. Therefore, it is necessary to use heuristic or metaheuristic methods to solve this problem in practical cases. In our study, the proposed metaheuristic is a Multi Start Iterative Local Search (MS-ILS), being this algorithm the most important contribution of this paper.

2 State of the Art

The multiple TRP (k-TRP) is a generelization for the TRP, where k routes must be determined, Jothi & Raghavachari (2007) and Fakcharoenphol *et al.* (2003) provide algorithms for approximation of this problem.

As it is described in Luo *et al.* (2014), the TRP has been extensively studied by a large number of researchers. This problem is also known as the travelling delivery problem (TDP) and the minimum latency problem (MLP) for their applications in different contexts. Some of the most recent researches of the TRP can be find in Jothi & Raghavachari (2007); Dewilde *et al.* (2013); Bock (2015), for the MLP the most recent researches are Wu *et al.* (2004); Silva *et al.* (2012); Angel-Bello *et al.* (2013); Lam *et al.* (2015) and for the TDP the latest research are Méndez-Díaz *et al.* (2008); Bjelić *et al.* (2013); Luo *et al.* (2014)

Multiple variations of the TRP are presented in the literature, including special constraints or features for activities or resources. For instance, the activities ($j \in V$) can include delivery limit time d_j (Polat *et al.* , 2015) or release times (Yu & Liu, 2009), and the resources (k), can consider limited capacity Q_k (Contardo & Martinelli, 2014) or congestion charge (Wen & Eglese, 2015).

3 Problem Definition and Mixed Integer Linear Model Formulation

Given a set of $J = (0, 1, ..., n, n+1)$ of $n+2$ customers and a set $R = (1, 2, ..., k)$ of k repairmen, each customer $j \in J$ must be assigned to a repairmen such that all the customers are performed by one repairmen. The repairmen cannot process customers simultaneously and the weighted sum of completion times at customers must be minimized. The customers 0 and $n+1$ respectively represent the depot and the end point of the distribution process.

The goal is to find the order of customers visit order that minimize the proposed objective function. In this case, the objective function is the weighted sum of completion times. This problem can be mathematically formulated by the following Mixed Integer Linear Program (MILP):

$$Min\ Z = \sum_{i=1}^{n} t_i \cdot w_i \tag{1}$$

$$\sum_{i=0}^{n+1} x_{ij} = 1, \qquad \forall\ j \in V \setminus \{0, n+1\} \tag{2}$$

$$\sum_{j=0}^{n+1} x_{ij} = 1, \qquad \forall\ i \in V \setminus \{0, n+1\} \tag{3}$$

$$\sum_{j=1}^{n} x_{0j} = k \tag{4}$$

$$t_j \geq t_i + s_{ij} + p_j - T \cdot (1 - x_{ij}), \qquad \forall\ i, j \in V \tag{5}$$

$$x_{ij} \in \{0, 1\}, \qquad \forall\ i, j \in V \tag{6}$$

$$t_j \geq 0, \qquad \forall\ j \in V \tag{7}$$

The binary decision variables x_{ij} indicate if the customer i is executed immediately after the customer j, and the decision variables t_j indicate the end time of the customer j; s_{ij} represents travel time between the customer i and customer j and p_j is the time to process the customer j, T is the required time for plan, design and complete the activities. All the s_{ij} and p_j values are assumed positives.

Equation (1) represents the objective function which minimize the weighted sum of completion times, the weights given by w_i can be interpreted as the importance of the customer i. Equations (2) and (3) indicate that each customer must be made exactly once, and each customer is scheduled before and after another customer, respectively. Equation (4) limits the repairmen and the Equation (5) restricted the completion time of each customer. Finally, Equations (6) and (7) define the decision variables domain.

4 Multi-Start Iterative Local Search Algorithm

The Multi-Start Iterative Local Search (MS-ILS) is an iterative approach in which every iteration perform an ILS metaheuristic which starts with a random initial solution and improves that solution by a process which include perturbation and local search procedures. The initial solutions is first enhanced by a local search heuristic. After that, an iterative process is started in which the resulting solution is disturbed several times and each perturbed solution is improved again by the local search heuristic. Each ILS iteration is over when the local search heuristic does not find any improvement, and them the best solution found is saved. Before the generation of a new initial solution, the best global solution is updated and returned at the end of the search process.

The MS-ILS is described by the following pseudo-code:

Algorithm 1 Pseudo-code of the proposed MS-ILS

```
Read (data, starts, perturbations)
Z(bestSol) = ∞
for starts do
  initialSol = Constructive()
  Precomputations(initialSol)
  sol = LS(initialSol)
  for perturbations do
    pertSol = Pert(sol)
    newSol = LS(pertSol)
    if Z(newSol) < Z(sol) then
      sol = newSol
    end if
  end for
  if Z(sol) < Z(bestSol) then
    bestSol = sol
  end if
end for
return bestSol
```

4.1 Initial solution

The construction of the initial solution is based on the nearest neighbor strategy. At the depot, the algorithm chooses the k closest customers. Each one of the k customers is assigned to a random vehicle as the first visited customer. Then the algorithm randomly assigns to each vehicle one of the two nearest customers from the last visited.

4.2 Precomputations

After an initial solution is obtained, the procedure *Precumputations* is performed to speed up the computation of the cost change by the subsequent local moves in the local search heuristic. This precomputations works as follows:

For each sequence of visited customers $\sigma = (1, 2, ..., |\sigma|)$, it is possible to define the values W_σ, D_σ and C_σ as the sum of weights, the total duration and the total cost or weighted sum of compeltion times of the sequence σ. Respectively these three values can be computed by Equations (8) to (10).

$$
\begin{aligned}
W_\sigma &= \sum_{i=1}^{|\sigma|} w_{\sigma(i)} && (8)\\
D_\sigma &= \sum_{i=1}^{|\sigma|-1} d_{\sigma(i),\sigma(i+1)} && (9)\\
C_\sigma &= \sum_{i=1}^{|\sigma|-1} \left(d_{\sigma(i),\sigma(i+1)} \cdot \sum_{j=i+1}^{|\sigma|} w_{\sigma(j)} \right) && (10)
\end{aligned}
$$

The concatenation operator $\oplus$ allows to compute the sum of weights, the total duration and the total cost of the sequence resulting from the combination of two sequences in order. This concatenation operator is based on Silva *et al.* (2012), who defined a concatenation operator for the sum of completion times objective function. Here we modified the operator of Silva et al. to be applied to the weighted sum of completion times.

Suppose there exist two sequences $A = (1, 2, 3)$ and $B = (4, 5)$. The sum of weights, the total duration and the total cost of sequences A and B are respectively, $W_A = w_1 + w_2 + w_3$, $W_B = w_4 + w_5$, $D_A = d_{12} + d_{23}$, $D_B = d_{23}$, $C_A = d_{12} \cdot (w_2 + w_3) + d_{23} \cdot w_3$ and $C_B = d_{45} \cdot w_5$. The concatenation of sequences A and B is defined as $A \oplus B = (1, 2, 3, 4, 5)$, and $W_{A\oplus B}$, $D_{A\oplus B}$, $C_{A\oplus B}$ can be computed by the following equations:

$$
\begin{aligned}
W_{A\oplus B} &= W_A + W_B && (11)\\
D_{A\oplus B} &= D_A + d_{34} + D_B && (12)\\
C_{A\oplus B} &= C_A + W_B \cdot (D_A + d_{34}) + C_B && (13)
\end{aligned}
$$

As every route in a solution is in fact a sequence of customers, the values of W, D and C for every sequence that can be extracted from a solution X are precomputed to speed up the metaheuristic. In practice we don't need to compute these values for all possible sequences. For each visited customer $i \in V$ by a route σ we precompute the arrays $\hat{W}_i$, $\hat{D}_i$ and $\hat{C}_i$ which correspond to the W, D and C values for the sequence $(\sigma_1, ..., \sigma_i)$. The values for each sequence starting and finishing in the customers in positions g and h of σ, $\hat{\sigma} = (\sigma_g, ..., \sigma_h)$, can be deduced in $O(1)$ from Equations (11) to (13) as follows:

$$
\begin{aligned}
W_{\hat{\sigma}} &= \hat{W}_h - \hat{W}_{g-1} && (14)\\
D_{\hat{\sigma}} &= \hat{D}_h - \hat{D}_g - d_{g-1,g} && (15)\\
C_{\hat{\sigma}} &= \hat{C}_h - \hat{C}_g - W_{\hat{\sigma}} \cdot (\hat{D}_g + d_{g-1,g}) && (16)
\end{aligned}
$$

4.3 Local Search

The local search uses unitary moves. This means that the algorithm evaluate moves which exchange one customer from a route k_1 with another one of a route k_2. The best improvement strategy is applied, that means when all possible changes are evaluated, the move with the best improvement is applied. The algorithm starts with the first customer of the first vehicle and evaluates if there is a possible exchange with another customer that improves the objective function. Notice there is not a restriction in the vehicles k_1 and k_2, therefore, the moves can involve the same vehicle ($k_1 = k_2$).

4.4 Perturbation

In order to explore other solutions and diversify the search space, the algorithm includes a perturbation procedure that consists in changing a random number of customers between two random chosen vehicles. The length of each exchanged sequence m is chosen by random. The sequence is selected such as it corresponds to the first m customers or the last m customer visited by the vehicles.

5 Computational Experiments

For the performance assessment of the MS-LS, seven instances of Christofides *et al.* (1979) for the capacitated VRP are used. The instances vary the number of customers and the repairmen as illustrated in Table 1.

Table 1: Instances set features

Instance	*1*	*2*	*3*	*4*	*5*	*11*	*12*
repairmen	5	10	8	12	17	7	10
customers	50	75	100	150	199	120	100

Before the execution of the MS-ILS for all the instances, it is necessary to define the number of starts and the number of perturbations that the metaheuristic must execute. Therefore, it is performed an experimentation over the first instance with different values for both parameters. The results are described in Table 2.

Table 2: Computational experiments with different parameter values

Type of instance	*Starts*	*Pert*	*Initial objective function value*	*Best objective function value*	*Improvement*	*Running time (sec)*
1	1	9999	3.82x10ˆ4	3,34x10ˆ4	4,81x10ˆ3	660
1	2	499	4,35x10ˆ4	3,35x10ˆ4	1x10ˆ4	593
1	5	199	4,08x10ˆ4	3,31x10ˆ4	7,64x10ˆ3	610
1	10	99	4,11x10ˆ4	3,39x10ˆ4	7,2x10ˆ3	684

Note that the times that the local search is applied is always the same (1000), so this results are used to determine that the best combinations of parameters is 5 starts with 199 perturbations and 2 starts with 499 perturbations. The criteria to choose those values are the best objective function values and the improvement when compared with initial solution objective function value. The idea with these combinations is to explore two scenarios: the first one with multiple initial solutions and the second one focused in the perturbations. Now, the obtained results with the seven instances are depicted in Tables 3 and 4.

Table 3: Results for seven instances with 5 starts and 199 perturbations

Type of instance	*Starts*	*Pert*	*Initial objective function*	*Best objective function*	*Improvement*	*Running time (sec)*
1	5	199	4.1x10ˆ4	3,35x10ˆ4	7,52x10ˆ3	590
2	5	199	5,16x10ˆ4	4,3x10ˆ4	8,57x10ˆ3	1372
3	5	199	7,52x10ˆ4	5,85x10ˆ4	1,67x10ˆ3	2952
4	5	199	1,03x10ˆ5	8,09x10ˆ3	2,22x10ˆ4	6954
5	5	199	1,24x10ˆ5	9,7x10ˆ4	2,69x10ˆ4	13268
6	5	199	1,57x10ˆ5	6,04x10ˆ4	9,69x10ˆ4	9375
7	5	199	1,07x10ˆ5	6,56x10ˆ4	4,17x10ˆ4	3174

Table 4: Results for seven instances with 2 starts and 499 perturbations

Type of instance	Starts	*Pert*	*Initial objective function*	Best objective function	Improvement	Running time (sec)
1	2	499	4.33x10ˆ4	3,23x10ˆ4	1,1x10ˆ4	581
2	2	499	5,46x10ˆ4	4,21x10ˆ4	1,25x10ˆ4	1136
3	2	499	7,07x10ˆ4	6,15x10ˆ4	9,25x10ˆ3	2594
4	2	499	9,34x10ˆ4	7,69x10ˆ4	1,65x10ˆ4	6639
5	2	499	1,14x10ˆ5	9,37x10ˆ4	2,03x10ˆ4	10.207
6	2	499	1,3x10ˆ5	6,64x10ˆ4	6,34x10ˆ4	7172
7	2	499	1,27x10ˆ5	6,48x10ˆ4	6,18x10ˆ4	2480

Figure 1 illustrates how is the change in the initial solution when a number of local search iterations are applied for the first instance of the set. The first point in the figure is the cost of the initial solution.

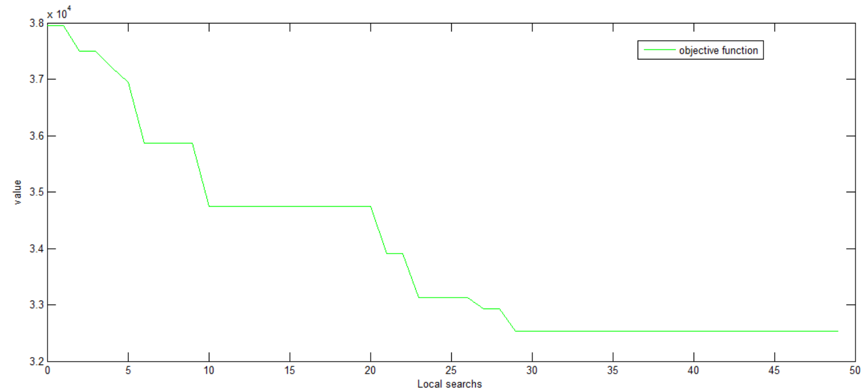

Figure 1: Improvement for different number of local search iterations

Figure 2 illustrates the metaheuristic elapsed times when different number of perturbations are applied. As in Figure 1, the instance used is the first one of the benchmark set. This figure shows a polynomial behaviour respect to the number of perturbations.

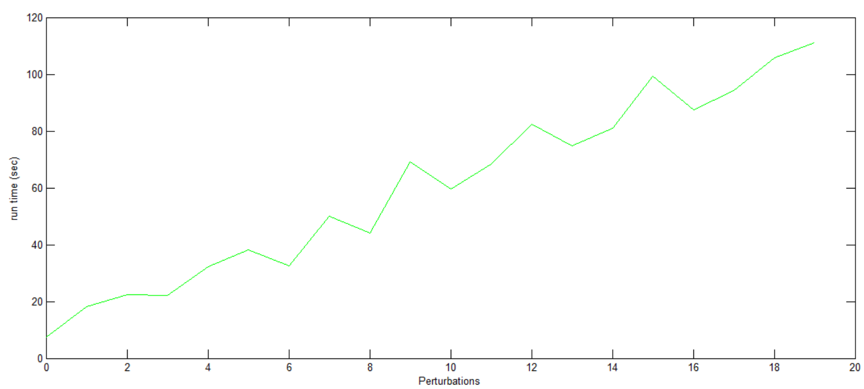

Figure 2: Running time for different number of perturbations

6 Conclusions and Further Work

Although the MS-LS is not compared with other heuristics, its performance seems to be good for solving vehicle routing problems because the improvements are relevant compared with the initial solution quality.

The obtained results are not determinant about the parameters behaviour. For instances 1,2, 3 and 7, the scenario with more initial solutions obtains better improvements. But for instances 4 to 6, the other scenario is better. To conclude if some of the parameters are more important is necessary to test more instances.

The running time of the heuristic seems to follow a lineal behavior. As it was expected, the problems with more repairmen and nodes, require more time to get the solutions. Note that although the number of LS applied are the same, the solution with 5 starts needs more time than the solution with 2 starts. It is explained because the first Local search that the algorithm executes after the initial solution requires more time because is when more changes are done in general. Local search procedures performed after perturbations require less computational effort to find an local optima solution.

Taking into account that parallel machine scheduling problems have the same mathematical formulation of the TRP (Joo & Kim, 2015), as further work, it is proposed to apply the MS-ILS to solve parallel machine scheduling problems.

References

Angel-Bello, Francisco, Alvarez, Ada, & García, Irma. 2013. Two improved formulations for the minimum latency problem. *Applied Mathematical Modelling*, **37**(4), 2257 – 2266.

Bjelić, Nenad, Vidović, Milorad, & Popović, Dražen. 2013. Variable neighborhood search algorithm for heterogeneous traveling repairmen problem with time windows. *Expert Systems with Applications*, **40**(15), 5997–6006.

Bock, Stefan. 2015. Solving the traveling repairman problem on a line with general processing times and deadlines. *European Journal of Operational Research*, **244**(3), 690 – 703.

Christofides, N., Mingozzi, A., & Toth, P. 1979. The vehicle routing problem. *Pages 315–338 of:* Christofides, N., Mingozzi, A., Toth, P., & Sandi, L. (eds), *Combinatorial optimization*. Wiley, Chichester.

Contardo, Claudio, & Martinelli, Rafael. 2014. A new exact algorithm for the multi-depot vehicle routing problem under capacity and route length constraints. *Discrete Optimization*, **12**(0), 129 – 146.

Dewilde, T., Cattrysse, D., Coene, S., Spieksma, F.C.R., & Vansteenwegen, P. 2013. Heuristics for the traveling repairman problem with profits. *Computers & Operations Research*, **40**(7), 1700–1707.

Fakcharoenphol, Jittat, Harrelson, Chris, & Rao, Satish. 2003. The K-traveling Repairman Problem. *Pages 655–664 of: Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '03. Society for Industrial and Applied Mathematics.

Joo, Cheol Min, & Kim, Byung Soo. 2015. Hybrid genetic algorithms with dispatching rules for unrelated parallel machine scheduling with setup time and production availability. *Computers & Industrial Engineering*, **85**(0), 102 – 109.

Jothi, Raja, & Raghavachari, Balaji. 2007. Approximating the k-traveling repairman problem with repairtimes. *Journal of Discrete Algorithms*, **5**(2), 293 – 303. 2004 Symposium on String Processing and Information Retrieval.

Lam, Nhat X., Tran, Tien, An, Min Kyung, & Huynh, Dung T. 2015. A note on the complexity of minimum latency data aggregation scheduling with uniform power in physical interference model. *Theoretical Computer Science*, **569**(0), 70 – 73.

Luo, Zhixing, Qin, Hu, & Lim, Andrew. 2014. Branch-and-price-and-cut for the multiple traveling repairman problem with distance constraints. *European Journal of Operational Research*, **234**(1), 49–60.

Méndez-Díaz, Isabel, Zabala, Paula, & Lucena, Abilio. 2008. A new formulation for the Traveling Deliveryman Problem. *Discrete Applied Mathematics*, **156**(17), 3223 – 3237. Cologne/Twente Workshop on Graphs and Combinatorial Optimization {CTW} 2005 Cologne/Twente Workshop on Graphs and Combinatorial Optimization.

Polat, Olcay, Kalayci, Can B., Kulak, Osman, & Günther, Hans-Otto. 2015. A perturbation based variable neighborhood search heuristic for solving the Vehicle Routing Problem with Simultaneous Pickup and Delivery with Time Limit. *European Journal of Operational Research*, **242**(2), 369 – 382.

Sahni, S., & Gonzalez, T. 1974. P-complete problems and approximate solutions. *Pages 28–32 of:* IEEE (ed), *15th annual symposium on switching and automata theory*.

Silva, Marcos Melo, Subramanian, Anand, Vidal, Thibaut, & Ochi, Luiz Satoru. 2012. A simple and effective metaheuristic for the Minimum Latency Problem. *European Journal of Operational Research*, **221**(3), 513 – 520.

Wen, Liang, & Eglese, Richard. 2015. Minimum cost {VRP} with time-dependent speed data and congestion charge. *Computers & Operations Research*, **56**(0), 41 – 50.

Wu, Bang Ye, Huang, Zheng-Nan, & Zhan, Fu-Jie. 2004. Exact algorithms for the minimum latency problem. *Information Processing Letters*, **92**(6), 303 – 309.

Yu, Wei, & Liu, Zhaohui. 2009. Vehicle routing problems on a line-shaped network with release time constraints. *Operations Research Letters*, **37**(2), 85 – 88.