OPTIMIZATION OF WATER SUPPLY VEHICLE ROUTING PROBLEM WITH TIME WINDOWS IN FLOOD DISASTER USING PARALLEL SIMULATED ANNEALING ALGORITHM

TERM PAPER

Fifi Alfiana Rosyidah*

Department of Industrial Engineering, Pusan National University
Busan, Korea

Abstract.
In emergency planning, routing decision to distribute the relief supply is an important decision. Especially clean water supply during and after a flood disaster. That problem is formulated as a vehicle routing problem with time windows. In this paper, the vehicle routing problem with time windows is developed using parallel simulated annealing algorithm. The solution refined from this research can help the decision maker to choose which route is better to reach the destination faster and optimized the vehicle amount.

Keywords: emergency planning, vehicle routing problem, parallel simulated annealing, metaheuristics, water supply chain, flood disaster

1. INTRODUCTION

The increase of population growth and high level of urbanization have increased the impacts of disasters, whether natural or man-made. Consequently, there is an increasing need for new methods to better prepare for and manage emergencies. In general, a good emergency response plan (ERP) must yield an efficient and immediate emergency response to locations that will be affected by the actual disasters. As a first stage of an ERP, in the preparation stage, emergency facilities must be sited at optimal locations, and the emergency vehicles (i.e. Fire Trucks, First Aid Vehicles etc.), that will be bringing relief material to emergency points from these facilities, be allocated to the right facilities. Later, during the actual emergencies, within short decision times, optimal routes for emergency vehicles must be found, and updated dynamically when new information about the emergencies becomes available (Oran *et al*. [1]).

Water may not be safe to drink, cook or clean after an emergency such as a flood. When flood disaster occurred, there often lack of clean water near the flooded area. Including the water in the Emergency Facility provided for the refugee. According to an environmental quality research by US government [2], during and after flooding, contaminated surface water may enter directly into the underground source of water. Water can become contaminated with microorganisms such as bacteria, sewage, heating oil, agricultural or industrial waste, chemicals and other substances that can cause serious illness. However, clean water is needed to supply to the refugee in each emergency facility.

During the supply chain of the clean water, the emergency planner in the emergency point should make a decision which route to take, in order to fulfill the delivery of the water to be on time and match the demands. This problem is formulated as Vehicle Routing Problem with Time Windows. To refining the solution of VRPTW, we present a fast and accurate metaheuristic that could update vehicle routes dynamically, and which could be utilized in real life emergencies where computation time is critical.

To serve a clean water to the refugees at the emergency facility, not only a certain demand that need to be fulfilled, but also the time windows has to be taken into consideration. Thus, a parallel simulated annealing algorithm based on Czezh and Czarnas [3] is used to solve the VRPTW in this paper. The objective is to find the best possible solutions to some well-known instances of the problem by using parallelism. The VRPTW considered in this work consist of finding a set of routes originating and terminating at an emergency point which serves a set of emergency facility where refugees is evacuate. For the purpose of delivery for clean water, there is a fleet of vehicles, each vehicle of some capacity. The emergency facility has the given demand of clean water and a vehicle on its route cannot serve more emergency facility that its capacity permit.

The organization of this paper is as follows. In section 2, the related work is presented. In section 3, the problem formulation and model development is formulated. In section 4 a numerical experiment is investigated. In section 5 conclusion and future research are given.

2. MOTIVATION AND RELATED WORK

According to Torre *et al*., disaster relief requires efforts on many fronts: providing rescue, health and medical assistance, water, food, shelter and long term recovery efforts. Much of successful and rapid relief relies on the logistical operations of supply delivery. In 2005, the United Nations established the Logistics Cluster as one of nine inter-agency coordination efforts in humanitarian assistance, recognizing the key importance of logistics in aid operations. The Pan American Health Organization (PAHO), a regional division of the World Health Organization (WHO), states in its publication Humanitarian Supply Management and Logistics in the Health Sector ([5]) that “countries and organizations must see [humanitarian supply logistics] as a cornerstone of emergency planning and preparedness efforts”.

Recently, many metaheuristics algorithm are used to solving several problems in the field of disaster management. Jahangiri *et al.* presented a model that determines optimum signal timing and increases the outbound capacity of the network using the simulated annealing algorithm which is a meta-heuristic technique. *Ma et al.* proposed the Genetic-Simulated Annealing Algorithm (GSAA) for solving location problem of emergency service facilities, which enriches searching behavior in optimization process, and has strong abilities of exploration and exploitation in large real search spaces. In this algorithm, genetic algorithm controls the search direction, and simulated annealing algorithm plays an important role in local optimum convergence. However, to our knowledge, there is no research work that solving vehicle routing problem using parallel simulated annealing in the field of disaster management.

3. PROBLEM FORMULATION AND MODEL DEVELOPMENT

This routing problem is an extension of the VRPTW formulation of Desrosiers et al. that will solve using parallel simulated annealing algorithms. In this paper, the parallel simulated annealing algorithm from Czezh and Czarnas is adopted. We assume there is set of facilities $J^k \subset J$ that house type-k vehicles, is already found for all $k \in K$.

Sets:

$I^k \subset I$: the et of emergency points that demand type-k vehicles,

$A^{j,k}$ $= \{(m,n) \in (I^k \cup \{j\}) \otimes (I^k \cup \{j\}) \mid \delta_{mj} \leq C,\ \delta_{nj} \leq C\}, j \in J^k, k \in K$

Parameters:

β_i^k : weight for serving type-k emergency at $i \in I^k$,

Δ_i^k : amount of clean water demanded from a type-k vehicle at $i \in I^k$,

D^k : distance capacity of type-k vehicles,

C^k : clean water capacity of type-k vehicles,

δ_{mn} : minimum distance between $m \in (I \cup J)$ and $n \in (I \cup J)$,

$[e_i, l_i]$: the time window at $i \in I$ within which emergency facility must be responded,

t_{mn}^k : travel time for a type-k vehicle on arc (m,n), $m \in (I \cup J)$ and $n \in (I \cup J)$,

s_i^k : service time for a type-k vehicle at $i \in I$, and $s_i^k = 0, \forall_i \in J$,

$t_{ij}^k = t_{mn}^k + s_i^k$.

Decision variables:

$$z_{mn}^{jk} = \begin{cases} 1 & \text{if the type-k vehicle housed at } j \in J^k \text{ travels on arc } (m,n) \in A^{j,k}, \\ 0 & \text{otherwise.} \end{cases}$$

$$T_n^k = \begin{cases} \text{Service start time at } n \text{ if } n \in I^k, \text{ for type-k vehicle,} \\ \text{Return time to facility } n \text{ if } n \in J^k, \text{ for type-k vehicle.} \end{cases}$$

The formulation:

$$\max z_{mn}^{jk} \sum_{k \in K} \sum_{j \in J^k} \sum_{(m,n) \in A^{j,k}} \beta_n^k\, z_{mn}^{jk}$$

Subject to:

$$\sum_{j \in J^k} \sum_{(m,n) \in A^{j,k}} z_{mn}^{jk} \leq 1, \qquad \forall k \in K, \forall m \in I^k, \tag{11}$$

$$\sum_{\{n \mid (m,n) \in A^{j,k}} z_{mn}^{jk} = \sum_{\{n \mid (m,n) \in A^{j,k}} z_{nm}^{jk}, \qquad \forall k \in K, \forall m \in (I^k \cup J^k), j \in J^k \tag{12}$$

$$\sum_{\{n \mid (j,n) \in A^{j,k}} z_{jn}^{jk} = 1, \qquad \forall k \in K, \forall j \in J^k, \tag{13}$$

$$\sum_{\{m \mid (m,j) \in A^{j,k}} z_{mj}^{jk} = 1, \qquad \forall k \in K, \forall j \in J^k, \tag{14}$$

$$z_{mj}^{jk} \left(T_m^k + t_{mn}^k - T_n^k\right) \leq 0, \qquad \forall k \in K, \forall j \in J^k, \forall (m,n) \in A^{j,k}, m \neq j \tag{15}$$

$$z_{jn}^{jk} \left(t_{jn}^k - T_n^k\right) \leq 0, \qquad \forall k \in K, \forall j \in J^k, \forall (j,n) \in A^{j,k} \tag{16}$$

$$e_m \leq T_m^k \leq l_m, \qquad \forall k \in K, \forall j \in J^k, \tag{17}$$

$$\sum_{(m,n)\in A^{j,k}} \delta_{mn}\ z_{mn}^{jk} < D^k \qquad \forall k \in K, \forall j \in J^k, \tag{18}$$

$$\sum_{(m,n)\in A^{j,k}} \Delta_n^k\, z_{mn}^{jk} < C^k \qquad \forall k \in K, \forall j \in J^k, \tag{19}$$

With that definition, the vehicles can only leave from and return to their own facilities. Also, they can only serve emergency points falling under the coverage area (distance C) of their facilities. Constraint (11) imposes that every emergency point can be assigned to at most one single route according to the type of emergency vehicle demanded at that point. (12), (13), (14) describe the flow for the type-k vehicle from facility j. (15), (16), and (17) ensure that vehicle routes stay feasible with the given time windows at each emergency facility. (18) guarantees feasibility for the range capacity, and (19) guarantees feasibility for the emergency clean water capacity of a vehicle.

4. ARCHITECTURE AND IMPLEMENTATION

The objective function is solved using parallel simulated annealing algorithm. The objective of using this metaheuristics algorithm is to achieve a high accuracy of solution to a problem. It is meant as its proximity to the global optimum solution. In the following is the parallel simulated annealing algorithm to solve VRPTW:

```
PROCESS P0:
Take the initial_solution as the best solution
  to the problem known so far;
Send initial_solution to processes Pj, j = 1, 2, ..., p;
final_solution := initial_solution;
equilibrium_counter := 0;
while equilibrium_counter ≤ τ do
  Receive best_local_solution_j from processes Pj,
    j = 1, 2, ..., p;
  Choose best_global_solution among
    best_local_solution_j, j = 1, 2, ..., p;
  if cost(best_global_solution) <
    cost(final_solution) then
    final_solution := best_global_solution;
    equilibrium_counter := 0; {final_solution
      was updated}
  else
    equilibrium_counter := equilibrium_counter + 1;
  end if;
  go := (equilibrium_counter ≤ τ);
  Send go to processes Pj, j = 1, 2, ..., p;
end while;
Produce final_solution as the solution to the VRPTW;

PROCESSES Pj, j = 1, 2, ..., p:
Receive initial_solution from P0;
old_solution_j := initial_solution;
best_local_solution_j := initial_solution;
T := γ * cost(initial_solution); {initial
  temperature of annealing}
loop
  for iteration_counter_j := 1 to n^2 do
```

```
        annealing_step(old_solution_j,
          best_local_solution_j); {as in the sequential
          algorithm}
        if number of annealing steps executed so far
          is a multiplicity of n then {co-operate}
          if j = 1 then Send best_local_solution_1
            to process P_2;
          else {j > 1}
            receive best_local_solution_{j-1}
              from process P_{j-1};
            if cost(best_local_solution_{j-1}) <
              cost(best_local_solution_j) then
              best_local_solution_j :=
                best_local_solution_{j-1};
            end if;
            if j < p then Send best_local_solution_j
              to process P_{j+1}; end if;
          end if;
        end if;
      end for;
      Send best_local_solution_j to P_0;
      Receive go from P_0;
      if go = false then stop; end if;
      T := β * T; {temperature reduction}
    end loop;
```

Compared to the standard simulated annealing algorithm, this extended algorithm proved effective to solve the delivery problem (Czech, 2001). Thus, we try to use this algorithm to our clean water supply vehicle routing problem.

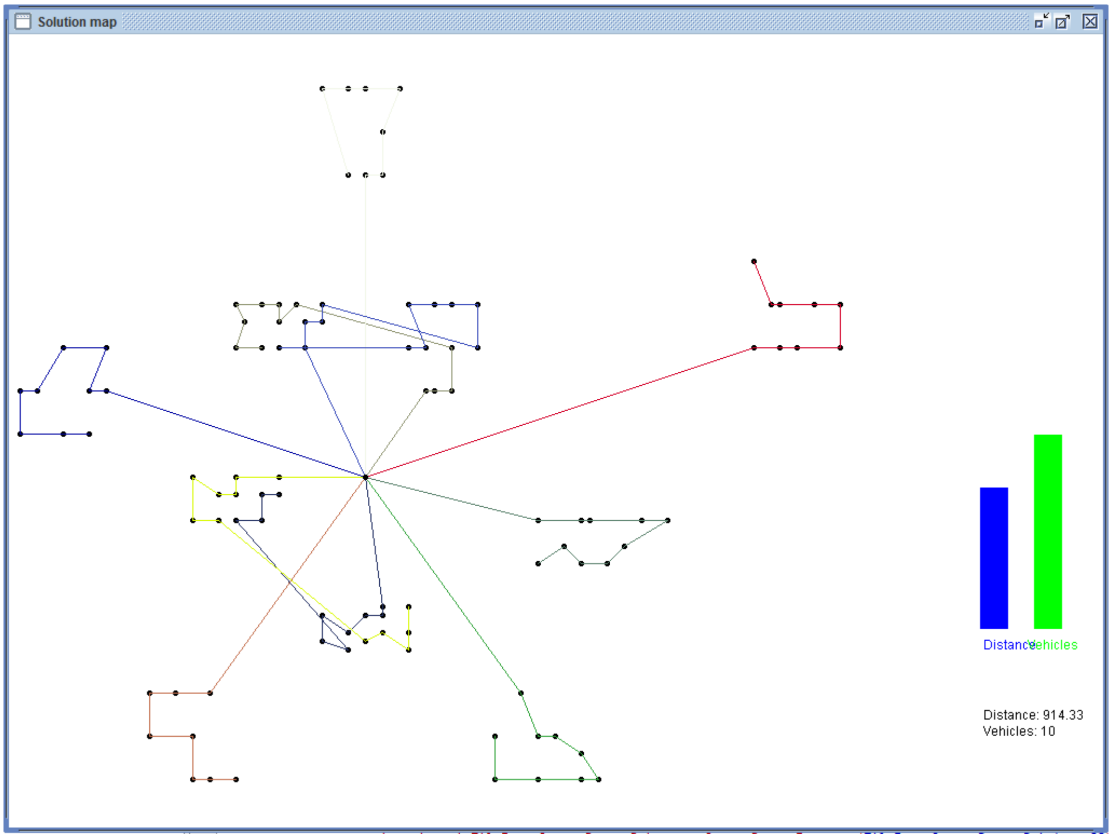
5. CASE STUDY

In this section, we treat some numerical examples and show the performance of the proposed algorithm. The numerical case was conducted by applying parallel simulated annealing algorithm which was coded in Java language programming. In the following is the example of the numerical data that we used:

Emergency Facility ID	Position X	Position Y	Demand	Start Time Window	Due Date Time	Service Time
1	40.00	50.00	0.00	0.00	1236.00	0.00
2	45.00	68.00	10.00	912.00	967.00	90.00
3	45.00	70.00	30.00	825.00	870.00	90.00
4	42.00	66.00	10.00	65.00	146.00	90.00
5	42.00	68.00	10.00	727.00	782.00	90.00
6	42.00	65.00	10.00	15.00	67.00	90.00

7	40.00	69.00	20.00	621.00	702.00	90.00
8	40.00	66.00	20.00	170.00	225.00	90.00
9	38.00	68.00	20.00	255.00	324.00	90.00
10	38.00	70.00	10.00	534.00	605.00	90.00

The data used in this paper is in total 101 data in the format as explained above. In the following is the result of the developed program:




```
run:
* INITIALIZE OPTIMIZATION *
Execution parameters: thread=8, non-improving=20, sigma=1.0, gamma=1.0, beta=0.92, delta=0.5
Starting solution: cost 6356.289182651073 (Distance = 3525.2891826510736, Total Vehicles = 28)
Found best solution cost ... 4236.064353148044 (Distance = 2714.0643531480437, Vehicles = 15)
Found best solution cost ... 4204.178732598832 (Distance = 2682.178732598832, Vehicles = 15)
Found best solution cost ... 4193.905151407689 (Distance = 2674.9051514076896, Vehicles = 15)
Found best solution cost ... 3921.050434886487 (Distance = 2399.050434886487, Vehicles = 15)
No improvement (1)
No improvement (2)
No improvement (3)
Found best solution cost ... 3833.1795326729643 (Distance = 2412.1795326729643, Vehicles = 14)
Found best solution cost ... 3631.3305380467746 (Distance = 2210.3305380467746, Vehicles = 14)
```

```
Found best solution cost ... 3549.4557818197686 (Distance = 2130.4557818197686, Vehicles = 14)
Found best solution cost ... 3314.376222824574 (Distance = 1994.3762228245741, Vehicles = 13)
Found best solution cost ... 3228.5525015357284 (Distance = 1908.5525015357284, Vehicles = 13)
No improvement (0)
Found best solution cost ... 3057.8276233542865 (Distance = 1738.8276233542867, Vehicles = 13)
No improvement (0)
No improvement (1)
```

The iteration will continue until the 20 no improvement happened before terminating and provide best solution.

8. CONCLUSIONS AND FUTURE WORK

In this paper, the clean water supply vehicle routing problem with time windows is solved using parallel simulated annealing algorithm. The solution refined from this research can help the decision maker to choose which route is better to reach the destination faster and optimized the vehicle amount. The developed program is still having several errors that might be need to be improved in the future.

REFERENCES

Idaho Department of Environmental Quality. After the Flood: Protecting Your Drinking Water. Accessed on 20 December 2016.

C. Ma, Y. Li, R. He, B. Qi, A. Diao. Research on location problem of emergency service facilities based on genetic-simulated annealing algorithm. Int. J. Wirel. Mobile Comput., 5 (2012), pp. 206–211

A. Jahangiri, S. Afandizadeh, N. Kalantari. The optimization of traffic signal timing for emergency evacuation using the simulated annealing algorithm. Transport, 26 (2011), pp. 133–140

Torre, Luis E de la, *et al*. Disaster relief routing: Integrating research and practice.