A NOTE ON NETWORK REPAIR CREW SCHEDULING AND ROUTING FOR EMERGENCY RELIEF DISTRIBUTION PROBLEM

HUAI-CHE HONG AND BERTRAND M. T. LIN*

Institute of Information Management
Department of Information Management and Finance
National Chiao Tung University
Hsinchu 300, Taiwan

(Communicated by Frank Werner)

Abstract. This paper proposes a dynamic programming algorithm for the NRCSRP with multiple crews. This algorithm also improves the existing algorithm for the problem with a single crew.

1. Introduction. The recent paper Duque, Dolinskaya, and Sörensen [2] considers a scheduling problem of a repair crew in emergency relief network. Among their interesting results and contributions, a dynamic programming algorithm is designed to produce optimal solutions. In this note, we propose an improved dynamic programming algorithm, which can be generalized to the case with multiple crews.

The problem, NRCSRP in short, is defined as followed. Let $G = (V, E)$ be an undirected connected graph with $V = V_d \cup V_r \cup \{0\}$, where set $E$ contains all undirected edges in the network and $V_d$ is the set of demand nodes (villages needing relief operations), $V_r$ is the set of damaged nodes to repair and 0 is the depot node where the relief supplies are positioned and from which the repair crew initially departs. Without loss of generality, each of the damaged roads is represented as a node in the middle of its corresponding link. For each village (node $i$), there is an importance factor $w_i$ that reflects the emergency level of its inhabitants. Each damage node $j$ represents the locations where the work of the repair crew is needed. Such nodes have a repair time $s_j$ that represents the time the repair crew spends on its first visit. Each edge $e_{i,j} \in E$ represents the link between nodes $i$ and $j$ in $V$. A travel time $t_{i,j}$ is defined for each edge $e_{i,j}$ to represent the travel time required by the repair crew to traverse on it. The goal of the NRCSRP is to find a plan for the crew to traverse the graph to let all villages accessible. The objective function to minimize the weighted sum of the time points each of the villages becomes accessible.

To devise a dynamic programming algorithm, Duque et al. [2] define states $(i, t, \bar{V}_r, \bar{V}_d)$, where node $i$ is the current location of the crew, $t$ is the current time, $\bar{V}_r$ is the set of damaged nodes not yet repaired, and $\bar{V}_d$ is the set of inaccessible villages. Let $n_d = |V_d|, n_r = |V_r|$ and $n = |V|$. Note that $t$ is a pseudo-polynomial parameter in the order of $T$, where $T = \sum_{i \in V} s_i + \sum_{i,j \in V, i \neq j} t_{i,j}$ and $\bar{V}_r$ and $\bar{V}_d$
are in the order of $O(2^n)$ and $O(2^{n^2})$, respectively. Therefore, the size of the state space is $O(n^2T2^n\cdot 2^n\nu)$. We elaborate on the relationship between $V_r$ and $V_d$ and find that the latter is derived when the former is given. The reachability of all nodes of $V_d$ can be determined by deploying a depth-first search starting from the depot in $O(|V| + |E|)$ time, which can be bounded from the above by $O(n^2)$. In other words, we can reduce the state space by eliminating the parameter $V_d$, which grows exponentially in the size of subsets $V_d$. This simplification will also lead to the development of a dynamic programming algorithm for multiple crews. To simplify the presentation, we introduce an algorithm for two crews which can be easily generalized for the case with any constant number of crews.

2. Dynamic programming algorithm for two crews. We adopt the notation of [2] for describing the new algorithm. Define state $(i_1, i_2, t_1, t_2, V_r)$ for the scenarios that satisfy the following two conditions

1. Amongst the damaged nodes that are already repaired, node $i_k$ is the last one crew $k$ completed and its completion time is $t_k$ for $k = \{1, 2\}$;
2. $V_r$ is the set of damaged nodes yet to be repaired at time $\max\{t_1, t_2\}$.

Each state describes the exact moment one or two nodes are being recovered. The scenario $t_1 < t_2$ indicates that the second crew recovers a node $v_2$ exactly at time $t_2$, while the first crew recovered another node $v_1$ at time $t_1$ and this crew is still on $v_1$ or en route for some other node. The scenario $t_1 > t_2$ indicates the reverse situation. If $t_1 = t_2$, then the two crews repair their nodes at the same time. Ranges of the state variables are $0 \leq i_1, i_2 \leq n + 1$, $V_r \subseteq V_r$, and $0 \leq t_1, t_2 \leq T$, where $T = \sum_{i \in V} s_i + \sum_{i, j \in V, i \neq j} t_{i, j}$. Define function $f(i_1, i_2, t_1, t_2, V_r)$ as the minimum cost incurred by moving from the initial state $s_0 = (0, 0, 0, 0, V_r)$ to state $s = (i_1, i_2, t_1, t_2, V_r)$. Let $V_d(i_1, i_2, t_1, t_2, V_r)$ denote the set of villages not yet rescued subject to the repair tasks at nodes of $V_r - V_r$ are completed. An optimal solution to the NRCSRP2 is the minimum value of $f(i_1, i_2, t_1, t_2, V_r)$ such that all demand nodes are rescued, i.e., the implied subset $V_d(i_1, i_2, t_1, t_2, V_r) = \emptyset$. A dummy node $v_0$ is introduced as the starting node for defining the initial condition. A backward dynamic program is then given as:

**Algorithm NRCSRP2**

**Initialization:**

$$f(i_1, i_2, t_1, t_2, V_r) = \begin{cases} 0, & \text{if } i_1 = i_2 = v_0, t_1 = t_2 = 0, V_r = V_r; \\ \infty, & \text{otherwise}. \end{cases}$$

**Recursion:**

For $i_1 \neq i_2 \in V_r$ and $V_r$:

$$f(i_1, i_2, t_1, t_2, V_r) = \begin{cases} \min_{i_1 \in V_r \setminus \{i_1, i_2\}} \left\{ w_{i_1}, t_1 + f(i_1, i_2, t_1, -t_{i_1, i_1}, (V_r \cup \{i_1\}), t_2, \right. & \text{if } t_1 > t_2, \left. \right\}, \right. \\
\min_{i_2 \in V_r \setminus \{i_1, i_2\}} \left\{ w_{i_2}, t_2 + f(i_1, i_2, t_1, t_2, -t_{i_2, i_2}, (V_r \cup \{i_2\}), \right. & \text{if } t_1 < t_2, \left. \right\}, \right. \\
\min \left\{ L, H \right\}, & \text{if } t_1 = t_2 \right. \right\}, \right.$$
where $\ell_{i_1, i_1}(\bar{V}_r \cup \{i_1\})$ is the minimum travel time from node $i_1'$ to node $i_1$ given that nodes of $\bar{V}_r \cup \{i_1\}$ are not yet repaired, and $\ell_{i_2', i_2}(\bar{V}_r \cup \{i_2\})$ is similarly defined. Furthermore,

\[
L = \min_{i_1' \in V_r \setminus \bar{V}_r \setminus \{i_1, i_2\}} \left\{ w_{i_2} t_2 + f(i_1, i_2', t_1, t_2 - \ell_{i_2', i_2}(\bar{V}_r \cup \{i_1\}), \bar{V}_r \cup \{i_1\} \cup \{i_2\}) \right\},
\]

\[
H = \min_{i_1' \in V_r \setminus \bar{V}_r \setminus \{i_1, i_2\}} \left\{ w_{i_1} t_1 + f(i_1', i_2, t_1, t_2 - \ell_{i_1', i_1}(\bar{V}_r \cup \{i_1\} \cup \{i_2\}), \bar{V}_r \cup \{i_1\} \cup \{i_2\}) \right\}.
\]

Goal: Find

\[
\min \left\{ f(i_1, i_2, t_1, t_2, \bar{V}_r) \mid \bar{V}_r \subseteq V_r, i_2 \in V_r \setminus \bar{V}_r, \bar{V}_d(i_1, i_2, t_1, t_2, \bar{V}_r) = \emptyset, 0 \leq t_1, t_2 \leq T \right\}.
\]

Regarding the running time of Algorithm NRCSRP2, we note that the state space has a size of $O(n^2 T^2 2^{2n_r})$. For each state, a procedure is invoked to find the shortest path. Using Fibonacci heaps, Dijkstra’s algorithm can be implemented in $O(n \log n + |E|)$ time, which is bounded from above by $O(n^2)$ [1]. Therefore, the overall running time is $O(n^2 n_r T^2 2^{2n_r})$. If the dynamic programming framework is generalized to $k$ crews, then complexity becomes $O(n^2 (n_r T)^k 2^{n_r})$.

3. Conclusions. This paper addressed a dynamic programming algorithm for the NRCSRP with multiple crews. This algorithm also improves on the existing algorithm for the problem with a single crew.

Acknowledgments. The authors are grateful to the anonymous referees for their constructive comments. This research was supported in part by the Ministry of Science and Technology of Taiwan under the grant MOST 106-2410-H-009-009.

REFERENCES


Received May 2017; 1st revision September 2017; 2nd revision November 2017.

E-mail address: edwardboy0627@gmail.com

E-mail address: bmtlin@mail.nctu.edu.tw