Network Disruption Recovery for Multiple Pairs of Shortest Paths

Lindong Liu
Department of IELM
The Hong Kong University of Science and Technology
Email: ldliu@ust.hk

Xiangtong Qi
Department of IELM
The Hong Kong University of Science and Technology
Email: ieemqi@ust.hk

***Abstract*—For an arc-disrupted network, we investigate the problem of partially recovering this network by given budget resource such that the total weighted transportation cost for all the origin-destination pairs is minimized. To obtain the solutions, we propose two heuristic algorithms based on Lagrangian relaxation, which both can generate good feasible solutions for the network disruption recovery problem. Accordingly, we are able to study how the marginal efficiency changes of providing additional resource and then decide the appropriate amount of budget resource to achieve the optimal social welfare.**

***Keywords*—network, transportation optimization, disruption recovery, Lagrangian relaxation**

I. Introduction

With the development of technology and economy, society is more and more demanding the reliability of various infrastructures such as transportation and telecommunication networks. However, disruptions do happen from time to time which damage the infrastructures, deteriorate services to users, and threat human lives. Such damages need to be repaired as quickly as possible, a process known as disruption recovery.

In the context of network disruption, both nodes and arcs may be disrupted. Here we consider the case of arc disruption. Specifically, a disrupted arc has an unusual high cost than its normal cost. Network disruption recovery is the problem for a set of origin-destination pairs. The central authority is to assign finite resource over some disrupted arcs to restore their functions and minimize the total transportation cost. One additional concern of the central authority is to decide the amount of budget resource used in recovery. This concern is of practical significance since higher budget resource, although leading to a lower total transportation cost, means a higher social opportunity cost.

In the last decade, on the research of disruption recovery management, different strategies have been proposed to fight the negative impact of unexpected disruptions through both proactive planning and aftermath rescheduling. The work is too extensive to review here. We refer to Clausen et al. (2010), Yu and Qi (2004), Kleindorfer and Saad (2005), Snyder and Daskin (2005) and Tomlin (2006) for some typical models and applications. Suffice it to say that, to the best of our knowledge, no previous work has specifically focused on the network arcs disruption recovery.

Lagrangian relaxation procedure is adopted here to solve the network recovery problem. Since first applied by Held and Karp (1970) to solve the travelling salesman problem, Lagrangian relaxation has been shown to be a very effective approach for solving integer linear programmings, such as scheduling problem in Muckstadt and Koenig (1977), assignment problem in Jörnsten and Näsberg (1986), vehicle routing problem in Kohl and Madsen (1997), facility location problem in Jain and Vazirani (2001) and lot sizing problem in Zhang et al. (2012), to name a few.

In this paper, we will develop two heuristic algorithms based on LR procedure and examine the effectiveness and efficiency by implementing them on large network recovery problems. Based on the feasible solutions given by the two heuristic algorithms, we then investigate the relationship between the total transportation cost and the amount of budget resource to check the marginal efficiency of providing additional resource.

The remainder of this paper is organised as follows. Section II describes the network recovery problem in a mathematical way and formulates this problem into an integer linear programming. Section III presents the Lagrangian relaxation framework and gives the details of the two Lagrangian relaxation based heuristic algorithms. Section IV reports the computational results.

II. Problem Description and Model

In a network recovery problem (NRP), there is an undirected disrupted network defined by $G = (N,E)$ with N being the set of nodes and E the set of disrupted arcs. Each arc $(i, j) \in E$ has a positive disrupted transportation cost c_{ij} and can be recovered to its normal transportation cost c'_{ij} with $0 \le c'_{ij} \le c_{ij}$ by consuming certain resource r_{ij}.

In total, there are m pairs of weighted origin-destination (OD) pairs. The goal of the decision maker is to minimize the total weighted transportation cost $\sum_{k=1}^{m} w^k \phi^k$ with budget resource R, where w^k is a positive weight and ϕ^k is the shortest distance to be determined for OD pair (o^k, d^k) after the recovery.

Apparently, more available resource will lead to lower total weighted transportation cost. However, in some cases, for the central authority, it may not pay to provide too much resource since, although the total weighted transportation cost is reduced, the social opportunity cost is increased. In this paper, besides provide heuristic algorithms to compute good feasible solutions for NRP, we will also investigate how $\sum_{k=1}^{m} w^k \phi^k$

978-1-4799-3134-7/14/$31.00 ©2014 IEEE

changes as R increases, and then try to give instructions of providing appropriate amount of resource in the view of central authority to maximize the social welfare, i.e., to minimize the summation of transportation and opportunity costs.

We formally list the notation used in NRP as follows.

TABLE I. NOTATION USED IN THE NETWORK RECOVERY PROBLEM

N :	The set of nodes, $N = \{1,2,...,n\}$.
M :	Indication set of OD pairs, $M = \{1,2,...,m\}$.
c_{ij} :	Disrupted unit transportation cost from node i to j.
c'_{ij} :	Normal unit transportation cost from node i to j.
r_{ij} :	Amount of resource consumed to recover arc (i,j).
o^k :	Origin of OD pair (o^k,d^k), $\forall k \in M$.
d^k :	Destination of OD pair (o^k,d^k), $\forall k \in M$.
w^k :	Weight of OD pair (o^k,d^k), $\forall k \in M$.
R :	Total available resource for the recovery.
x^k_{ij} :	Decision variable. $x^k_{ij} = 1$, if arc (i,j) is a recovered arc on the shortest path of pair (o^k,d^k) and $x^k_{ij} = 0$, otherwise, $\forall (i,j) \in E, k \in M$.
y^k_{ij} :	Decision variable. $y^k_{ij} = 1$, if arc (i,j) is a disrupted arc on the shortest path of pair (o^k,d^k) and $y^k_{ij} = 0$, otherwise, $\forall (i,j) \in E, k \in M$.
z_{ij} :	Decision variable. $z_{ij} = 1$, if arc (i,j) is recovered and $z_{ij} = 0$, otherwise, $\forall (i,j) \in E, k \in M$.

The network recovery problem can be written as the following integer linear programming (P),

$$\Omega_P := \min \sum_{k \in M} w^k \sum_{(i,j) \in E} \left(c'_{ij} x^k_{ij} + c_{ij} y^k_{ij} \right) \tag{1}$$

$$s.t. \quad \sum_{j:(i,j) \in E} \left(x^k_{ij} + y^k_{ij} \right) - \sum_{j:(j,i) \in E} \left(x^k_{ji} + y^k_{ji} \right)$$

$$= \begin{cases} 1, & \text{if } i = o^k \\ -1, & \text{if } i = d^k \\ 0, & \text{if } i \notin \{o^k, d^k\} \end{cases} \quad \forall i \in N, k \in M \tag{2}$$

$$\sum_{(i,j) \in E} r_{ij} z_{ij} \leq R \tag{3}$$

$$x^k_{ij} \leq z_{ij}, \ \forall (i,j) \in E, k \in M \tag{4}$$

$$x^k_{ij}, y^k_{ij}, z_{ij} \in \{0,1\}, \ \forall (i,j) \in E, k \in M. \tag{5}$$

In the above formulations, (1) is the objective to minimize the total weighted transportation cost for the central authority; (2) are the flow conservation constraints which identify a unique path from origin o^k to destination d^k for each OD pair (o^k,d^k); (3) is budget balance constraint which means the total consumed resource $\sum_{(i,j) \in E} r_{ij} z_{ij}$ cannot exceed budget R; constraints (4) ensure the fact that only if an arc (i,j) is recovered can this arc be used as a normal arc on some shortest paths; constraints (5) are binary requirements for the decision variables.

The network recovery problem is hard to solve directly, since constraint (3) itself already leads to a knapsack problem which is pseudo polynomial solvable.

III. HEURISTIC ALGORITHMS FOR NRP

In this section, we will develop two heuristic algorithms, based on Lagrangian relaxation procedure, to obtain good feasible solutions for the network recovery problem in reasonable computational time.

A. The Lagrangian Relaxation Procedure

Based on the formulation of NRP, we relax constraints (4) to the objective function with Lagrangian multiplier λ. The resulting Lagrangian function (LF) is,

$$\begin{aligned} L(\lambda) = & \min_{\left\{x^k_{ij}, y^k_{ij}, z_{ij}\right\}} \sum_{k \in M} w^k \sum_{(i,j) \in E} \left(c'_{ij} x^k_{ij} + c_{ij} y^k_{ij} \right) \\ & + \sum_{k \in M} \sum_{(i,j) \in E} \lambda^k_{ij} \left(x^k_{ij} - z_{ij} \right) \\ & \text{subject to } (2),(3) \textit{ and } (5) \\ = & \min_{\left\{x^k_{ij}, y^k_{ij}, z_{ij}\right\}} \sum_{k \in M} w^k \sum_{(i,j) \in E} \left[\left(c'_{ij} + \frac{\lambda^k_{ij}}{w^k} \right) x^k_{ij} + c_{ij} y^k_{ij} \right] \\ & - \sum_{k \in M} \sum_{(i,j) \in E} \lambda^k_{ij} z_{ij} \\ & \text{subject to } (2),(3) \textit{ and } (5). \end{aligned} \tag{6}$$

According to Lagrangian duality theory, for any non-negative Lagrangian multiplier λ, $L(\lambda)$ gives a lower bound on its primal optimal objective value Ω_P. To obtain the sharpest lower bound, we need to optimize the corresponding Lagrangian dual problem (LDP) given by,

$$\begin{aligned} L^* = & \max_{\lambda \geq 0} L(\lambda) \\ & \text{subject to } (2),(3) \textit{ and } (5). \end{aligned} \tag{7}$$

Before solving (7), we first need to give a way of computing $L(\lambda)$. In ILP (6), since constraints (2) and (3) are independent, Lagrangian function can be decomposed into a shortest path problem (SP) with optimal objective value $L_s(\lambda)$ and a knapsack problem (KP) with optimal objective value $L_k(\lambda)$ such that $L(\lambda) = L_s(\lambda) + L_k(\lambda)$.

The specific expression of the shortest path problem is

$$\begin{aligned} & L_s(\lambda) := \min \sum_{k \in M} w^k \sum_{(i,j) \in E} \left(c''_{ij} x^k_{ij} + c_{ij} y^k_{ij} \right) \\ & s.t. \sum_{j:(i,j) \in E} \left(x^k_{ij} + y^k_{ij} \right) - \sum_{j:(j,i) \in E} \left(x^k_{ji} + y^k_{ji} \right) \\ & = \begin{cases} 1, & \text{if } i = o^k \\ -1, & \text{if } i = d^k \\ 0, & \text{if } i \notin \{o^k, d^k\} \end{cases} \quad \forall i \in N, k \in M \\ & \quad x^k_{ij}, y^k_{ij} \in \{0,1\}, \ \forall (i,j) \in E, k \in M. \end{aligned} \tag{8}$$

where $c''_{ij} = c'_{ij} + \frac{\lambda^k_{ij}}{w^k}$.

The SP (8) can be solved in polynomial time by, first, choosing the smaller value between c''_{ij} and c_{ij} as the new transportation cost, for each arc $(i,j) \in E$, and second, using Dijkstra's algorithm (see, Skiena 1990) to obtain the shortest distance $l^k(\lambda)$ for each OD pair (o^k,d^k). Then, $L(\lambda) = \sum_{k \in M} l^k(\lambda)$.

The specific expression of the knapsack problem is

$$
\begin{aligned}
L_k(\lambda) := \min - \sum_{k\in M}\sum_{(i,j)\in E} \lambda_{ij}^{k} z_{ij} \\
s.t. \sum_{(i,j)\in E} r_{ij} z_{ij} \le R \\
z_{ij} \in \{0,1\},\ \forall (i,j)\in E.
\end{aligned} \tag{9}
$$

where $\lambda_{ij} = \sum_{k\in M}\lambda_{ij}^{k}$.

ILP (9) is a standard knapsack problem where there is a bag with capacity R and a set of items $\{(i,j) : \forall (i,j) \in E\}$ each with volume r_{ij} and profit λ_{ij}. Therefore, this knapsack problem can be solved in pseudo polynomial time by dynamic programming (see, Andonov et al. 2000).

Note that the optimal solution z_k for (9) is, although not optimal, a feasible resource assignment solution for the network recovery problem. We will utilize z_k to develop good feasible solutions for NRP, the detailed operations will be described in the design of Lagrangian heuristic algorithms.

To find the optimal Lagrangian multipliers λ^* that maximizes $L(\lambda)$, we apply subgradient method (see, Held et al. 1974) to update Lagrangian multiplier λ at each iteration. The recursive formula is, for any $(i,j) \in E, k \in M$,

$$
\lambda_{ij}^{k(t+1)} = \left\{ \lambda_{ij}^{k(t)} + \theta^{(t)} \left[x_{ij}^{k(t)} - z_{ij}^{(t)} \right] \right\}^{+}. \tag{10}
$$

In (10), notation $\{a\}^+$ denotes the positive part of a, that is, $\{a\}^+ = \max\{0,a\}$; $\lambda^{(t)}$ and $\lambda^{(t+1)}$ represent the values of Lagrangian multiplier λ at iterations t and $t+1$, respectively; vector $\left[x^{1(t)} - z^{(t)}, x^{2(t)} - z^{(t)}, \ldots, x^{m_p(t)} - z^{(t)}\right]$ indicates the gradient at point $\left(\lambda^{(t)}, L(\lambda^{(t)})\right)$; $\theta^{(t)}$ is the step size at iteration t given by,

$$
\theta^{(t)} = \frac{\rho^{(t)} \left[UB - L(\lambda^{(t)}) \right]}{\sum_{k=1}^{m} \| x^{k(t)} - z^{(t)} \|^2},
$$

where UB is an upper bound on Ω_P, i.e., the smallest objective value of P found by iteration t; $L(\lambda^{(t)})$ is the Lagrangian function value at iteration t which is equal to $L_s(\lambda^{(t)}) + L_k(\lambda^{(t)})$; $\rho^{(t)}$ is a scalar initially set to 2 and reduced to half whenever the maximal Lagrangian function value $L(\tilde{\lambda})$ found so far has failed to increase in a specified number of iterations, or a larger $L(\tilde{\lambda})$ is found.

In general, in a standard Lagrangian relaxation procedure, at each iteration t, we put in Lagrangian multipliers $\lambda^{(t)}$ derived by subgradient method; after solving two sub-problems SP and KP, we can generate a feasible resource assignment solution $z^{(t)}$ for the network recovery problem and compute the corresponding objective value to update UB; recursively derive Lagrangian multipliers $\lambda^{(t+1)}$. Repeat this process until the gap between UB and the maximal Lagrangian function value $L(\tilde{\lambda})$ is close to zero or the number of iterations reaches to a limit.

B. The Lagrangian Relaxation Heuristics

According to the Lagrangian relaxation procedure, for the network recovery problem, we can generate a good lower bound $L(\tilde{\lambda})$ called Lagrangian lower bound (LRB) from standard LR procedure. A by-product of this procedure is a resource assignment solution, obtained by solving the knapsack problem, which is feasible for NRP.

To be specific, in LR procedure, the iterative solution $\left[x^{(t)}; y^{(t)}; z^{(t)}\right]$ for Lagrangian function might not be feasible for network recovery problem due to the relaxation of constraints $x_{ij}^{k} \le z_{ij}$, however, $z^{(t)}$ itself is its feasible resource assignment solution. Unfortunately, $z^{(t)}$ is not necessarily reasonable, since in some situations, arcs are recovered but not used in shortest paths of any OD pairs. To this end, we will develop two heuristic algorithms to improve $z^{(t)}$ and obtain better feasible solutions for the network recovery problem.

Algorithm 1. *Lagrangian heuristic algorithm one: in the t^{th} iteration of Lagrangian relaxation procedure, based on $z^{(t)}$, an improved resource assignment solution $z'^{(t)}$ is developed as follows.*

Step 1*: According to solution $z^{(t)}$, recover the network, and determine the shortest path $SP_k^{(t)}$ for each OD pair (o^k, d^k). Denote the shortest paths set as $SP^{(t)} \equiv \left\{SP_1^{(t)}, SP_2^{(t)}, \ldots, SP_m^{(t)}\right\}$.*

Step 2*: Among all the recovered arcs, some appear in $SP^{(t)}$ and some do not. Select the arcs which do not appear in any shortest paths and release the corresponding resource consumed in recovery. Denote the selected arcs as $\tilde{z}^{(t)}$ and the total released resource as $R'^{(t)}$.*

Step 3*: Among all the arcs which appear in shortest paths set $SP^{(t)}$, denote the unrecovered arcs as E', recover some arcs in E' with resource $R'^{(t)}$ such that the total weighted transportation cost is maximally reduced by following shortest paths set $SP^{(t)}$. We call the newly recovered arcs $\bar{z}^{(t)}$ the complemented resource assignment solution. The improved solution $z'^{(t)}$ is equal to $z^{(t)} - \tilde{z}^{(t)} + \bar{z}^{(t)}$.*

In Algorithm 1, the shortest paths in step 1 are determined by Dijkstra's algorithm. Finding the complemented resource assignment solution $\bar{z}^{(t)}$ is equivalent to solving a knapsack problem where there is a bag with capacity $R'^{(t)}$ and a set of items $\{(i,j) : (i,j) \in E'\}$ each with volume r_{ij} and profit $\sum_{k=1}^{m} w^k \gamma_{ij}^{k} (c_{ij} - c'_{ij})$. Here, γ_{ij}^{k} is a binary incidence parameter which is equal to 1 if arc (i,j) is on path $SP_k^{(t)}$ and 0, otherwise.

After Lagrangian relaxation procedure, we choose a resource assignment solution z'^* with the minimal total weighted transportation cost among the iterative solutions $z'^{(t)}$ as the final solution. Denote the sharpest Lagrangian lower bound and the smallest total weighted transportation cost found in Al

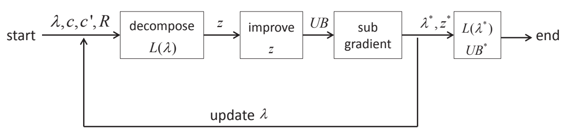

Fig. 1. Lagrangian heuristic one

In the first heuristic algorithm, we improved iterative solution $z^{(t)}$ to $z'^{(t)}$ at each LR iteration, and obtain final solution z'^{*} out of all iterative solutions $z'^{(t)}$. In the second heuristic algorithm, rather than release resource at each iteration, we release the wasted resource of the best iterative solution z^* among $z^{(t)}$ after a complete LR procedure. Instead of using the released resource to solve a knapsack problem, we turn to solve a reduced network recovery problem with updated parameters R, c_{ij}, and c'_{ij}. The detailed descriptions of this algorithm are as follows.

Algorithm 2. *Lagrangian heuristic algorithm two.*

Step 1*: After a complete Lagrangian relaxation procedure, according to the best resource assignment solution z^* selected from $z^{(t)}$ given at each iteration, recover arcs $\widehat{z}$ appearing in the corresponding shortest paths set SP^* of z^* and obtain the released resource R' by using similar method described in steps 1 and 2 of the first heuristic.*

Step 2*: For each arc $(i,j) \in E$, if it is recovered, let $c_{ij} = c'_{ij}$. Under the new settings, treat the original problem P as a reduced netwrok recovery problem with total resource R' and resolve this problem.*

Step 3*: Repeat steps 1 and 2 until we find an enough good solution or the released resource R' is no longer helpful to reduce the transportation cost. The final recovery decision is just the summation of all best resource assignment solutions z^* for each problem solved in steps 1 and 2.*

Denote the sharpest Lagrangian lower bound and the s-

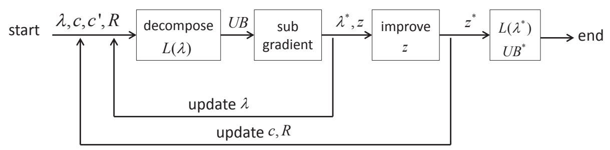

Fig. 2. Lagrangian heuristic two

In general, both of the two heuristic algorithms can generate good feasible solutions for the network recovery problem in reasonable computing time. We will show the computational results in next section.

IV. Computational Results

We conduct all computational experiments on a Windows 7 PC with an Intel Core i7-2600 and 16G RAM, 3.4GHz CPU. All algorithms are implemented by Matlab 2011a.

We now show the effectiveness and efficiency of the proposed Lagrangian heuristic algorithms. In the experiments, we randomly construct 3 types of networks which all contain 50 OD pairs and 100 nodes while have different densities, to be specific, they contain 500, 1500 and 4500 undirected arcs, respectively. The normal transportation costs $\left\{c'_{ij} : \forall (i,j) \in E\right\}$ are integers uniformly distributed in interval $\left[10,30\right]$. The disrupted transportation cost c_{ij} is equal to the product of c'_{ij} and a real value uniformly distributed in interval $\left[1,2\right]$. The amount of resource r_{ij} to recover arc (i,j) is an integer uniformly distributed in interval $[10,30]$. The weights of OD pairs w are integers uniformly distributed in interval $\left[5,15\right]$. We analyse the results where the total available resource are 500, 1000 and 1500, respectively. In other words, the total available resource can recover 25, 50 and 75 arcs, respectively, on average.

The detailed results are presented in Tables II, III and IV.

TABLE II. NRP WITH 100 NODES AND 500 UNDIRECTED ARCS

R	Heuristic one (%)			Heuristic two (%)		
	Avg.	Max.	Min.	Avg.	Max.	Min.
500	86.42	89.30	82.51	86.25	88.05	84.07
1000	89.78	91.67	87.46	90.03	91.95	87.88
1500	95.96	98.06	93.36	96.12	98.16	93.54

R	Heuristic (%)			T_1 (s)	T_2 (s)
	Avg.	Max.	Min.		
500	86.89	89.47	84.07	1416	1325
1000	90.04	91.95	87.88	1319	1311
1500	96.12	98.16	93.54	1305	1303

TABLE III. NRP WITH 100 NODES AND 1500 UNDIRECTED ARCS

R	Heuristic one (%)			Heuristic two (%)		
	Avg.	Max.	Min.	Avg.	Max.	Min.
500	86.92	89.43	84.52	86.80	88.62	84.22
1000	92.65	94.51	90.49	92.76	94.70	90.58
1500	98.66	99.79	97.13	98.73	99.79	97.40

R	Heuristic (%)			T_1 (s)	T_2 (s)
	Avg.	Max.	Min.		
500	87.22	89.61	84.52	1346	1327
1000	92.76	94.70	90.59	1353	1432
1500	98.73	99.79	97.40	1313	1306

TABLE IV. NRP WITH 100 NODES AND 4500 UNDIRECTED ARCS

R	Heuristic one (%)			Heuristic two (%)		
	Avg.	Max.	Min.	Avg.	Max.	Min.
500	87.08	89.00	84.91	87.21	89.35	85.35
1000	95.87	99.19	93.86	96.00	99.36	94.06
1500	99.93	100	98.85	99.94	100	99.08

R	Heuristic (%)			T_1 (s)	T_2 (s)
	Avg.	Max.	Min.		
500	87.39	89.36	85.61	1259	1264
1000	96.00	99.36	94.06	1247	1250
1500	99.94	100	99.08	291	1253

In these tables, columns "T_1" and "T_2" separately represent the computational time of carrying out Lagrangian heuristic algorithms one and two on the network recovery problem. The values under column "Heuristic one", "Heuristic two" and "Heuristic" are the percentage ratios of LRB_1 over Ω_1, LRB_2 over Ω_2 and $LRB^* = \max\left\{LRB_1, LRB_2\right\}$ over $\Omega^* = \min\left\{\Omega_1, \Omega_2\right\}$, respectively. From the above computational results, we have the following observations.

First, both Lagrangian heuristic algorithms are effective and efficient in solving network recovery problem. The average effectivenesses are larger than 85% under all situations. All problems can be solved within 25 minutes.

Second, both algorithms become more effective when the budget resource is increasing. As the budget resource is large enough, we even can find the optimal solution for the network recovery problem which leads to the corresponding percentage ratio being 100%.

Third, the first heuristic algorithm is more effective when the total resource is small while the second one can do better when the total resource is relatively large. These two algorithm can be good alternatives for each other when solving network recovery problem.

Forth, as the network is denser, both heuristic algorithms become more effective. This in turn shows the good applicability of our heuristic algorithms on larger and denser networks.

In Tables II, III and IV, we have shown the effectivenesses and efficiencies of our two Lagrangian heuristic algorithms implemented under various situations for network recovery problem. However, for the central authority, the main concern also includes deciding the amount of resource, when facing a particular disrupted network, such that the social welfare is maximized. To this end, we construct 20 different disrupted networks, each with 100 nodes and 1500 undirected arcs. For each network, the parameter settings are the same as those introduced at the beginning of this section, the only difference is that the available resource here is a series of integer values from 100 to 2000 with interval 100. We present the computational results, called "R-Ω^* curve", in the following diagram where the horizontal and vertical axes represent the input amount of resource and the average best found total weighted transportation cost, respectively, for the 20 instances.

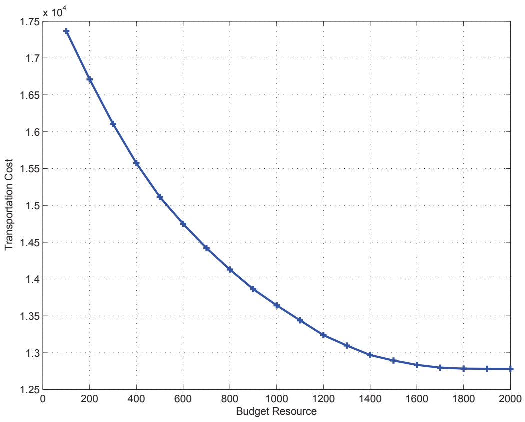

Fig. 3. Average R-Ω^* curve of 20 NRPs

According to Figure 3, denote the 20 highlighted sampling points as $\{s_1, s_2, \cdots, s_{20}\}$ with coordinates $\{(R_1, \Omega_1^*), (R_2, \Omega_2^*), \cdots, (R_{20}, \Omega_{20}^*)\}$. For any $k \in \{2, 3, \cdots, 20\}$, let the marginal efficiency at R_k be equal to $\Delta\Omega_k^* = \Omega_k^* - \Omega_{k-1}^*$ which means the reduced transportation cost by additionally providing 100 units of budget resource. Table V shows the detailed marginal efficiency at each sampling point.

TABLE V. NRP WITH 100 NODES AND 4500 UNDIRECTED ARCS

R	100	200	300	400	500	600	700
$\Delta\Omega^*$	N.A.	655	603	533	457	366	330
R	800	900	1000	1100	1200	1300	1400
$\Delta\Omega^*$	290	266	220	204	202	140	129
R	1500	1600	1700	1800	1900	2000	-
$\Delta\Omega^*$	74	60	37	14	2	0	-

Based on Figure 3 and Table V, we can conclude that the marginal efficiency of providing additional budget resource is decreasing for the network recovery problem. When the budget resource is high enough, e.g., larger than 2000 in our case, the marginal efficiency even drops to 0 which indicates a potential waste of providing more resource. If given the exact marginal social opportunity cost of the budget resource, the central authority can decide the amount of resource used to recover the network.

V. Conclusion

The focus of this paper is on the network recovery problem. We have proposed two Lagrangian relaxation based heuristic algorithms which, as shown in the computational results part, both can generate good feasible solutions for network recovery problem. Besides, we have also provided the "R-Ω^* curve" to help the central authority decide the appropriate amount of budget resource.

An interesting extension of this paper is taking the recovery time into consideration. Usually disruption recovery takes certain duration of time, e.g., several days or even longer, before the arcs can fully recover. Meanwhile, partially recovered arcs can provide partial services to users. Different sequence of recovery will lead to different service levels to users during the recovery process. Further study can focus on the problem that how a disrupted network can be gradually recovered during the transition period of recovery.

References

Andonov, R., V. Poirriez, and S. Rajopadhye (2000). Unbounded knapsack problem: Dynamic programming revisited. *European Journal of Operational Research 123*(2), 394–407.

Clausen, J., A. Larsen, J. Larsen, and N. J. Rezanova (2010). Disruption management in the airline industryłconcepts, models and methods. *Computers & Operations Research 37*(5), 809–821.

Held, M. and R. M. Karp (1970). The traveling-salesman problem and minimum spanning trees. *Operations Research 18*(6), 1138–1162.

Held, M., P. Wolfe, and H. P. Crowder (1974). Validation of subgradient optimization. *Mathematical Programming 6*(1), 62–88.

Jain, K. and V. V. Vazirani (2001). Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. *Journal of the ACM (JACM) 48*(2), 274–296.

Jörnsten, K. and M. Näsberg (1986). A new lagrangian relaxation approach to the generalized assignment problem. *European Journal of Operational Research 27*(3), 313–323.

Kleindorfer, P. R. and G. H. Saad (2005). Managing disruption risks in supply chains. *Production and Operations Management 14*(1), 53–68.

Kohl, N. and O. B. Madsen (1997). An optimization algorithm for the vehicle routing problem with time windows based on lagrangian relaxation. *Operations Research 45*(3), 395–406.

Muckstadt, J. A. and S. A. Koenig (1977). An application of lagrangian relaxation to scheduling in power-generation systems. *Operations Research 25*(3), 387–403.

Skiena, S. (1990). Dijkstra's algorithm. *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica, Reading, MA: Addison-Wesley*, 225–227.

Snyder, L. V. and M. S. Daskin (2005). Reliability models for facility location: the expected failure cost case. *Transportation Science 39*(3), 400–416.

Tomlin, B. (2006). On the value of mitigation and contingency strategies for managing supply chain disruption risks. *Management Science 52*(5), 639–657.

Yu, G. and X. Qi (2004). *Disruption management*. World Scientific.

Zhang, Z.-H., H. Jiang, and X. Pan (2012). A lagrangian relaxation based approach for the capacitated lot sizing problem in closed-loop supply chain. *International Journal of Production Economics 140*(1), 249–255.