Using the Lagrange Relaxation Method to Solve the Routing and Packet Scheduling Problem in IEEE 802.16 Mesh Networks

Meng-Hao Liu
Department of Electrical Engineering
National Yunlin University of Science and Technology
Taiwan, ROC
Email: g9612716@yuntech.edu.tw

Pi-Rong Sheu
Department of Electrical Engineering
National Yunlin University of Science and Technology
Taiwan, ROC
Email: sheupr@yuntech.edu.tw

Chi-Chiuan Liou
Department of Electrical Engineering
National Yunlin University of Science and Technology
Taiwan, ROC
Email: liouchi@yuntech.edu.tw

Ren-Hao Liao
Department of Electrical Engineering
National Yunlin University of Science and Technology
Taiwan, ROC
Email: g9812719@yuntech.edu.tw

***Abstract*—In an IEEE 802.16 mesh network, the routing and packet scheduling (RPS) problem is to design a fast scheduling scheme to meet the requirement of all the subscriber stations (SSs) so that all of the packets can be delivered to the base stations (BS) while minimizing the number of time slots and prohibiting the interference between any two packets. There has existed an integer linear programming formulation (ILPF) for the RPS problem, named as ILPF-for-RPS, which can find its optimal solution. However, the execution time of MIPF-for-MSC is very long when the number of SSs is large. This is because that the RPS problem has been proven to be NP-complete. In this paper, the Lagrange relaxation method is applied to shrink the solution space of the ILPF-for-RPS for the purpose of shortening the solution-searching time. First, we theoretically transform the ILPF-for-RPS into a Lagrange relaxation ILPF-for-RPS, whose objective function is then proved to be a concave function. This Lagrange relaxation ILPF-for-RPS simplifies the original ILPF-for-RPS and can be used to find an optimal solution for the RPS problem within a shorter time. According to the computer simulations, in contrast to the original ILPF-for-RPS, our Lagrange relaxation ILPF-for-RPS can attain a minimum time slot schedule within a shorter time for the RPS problem. To be more specific, compared with the original ILPF-for-RPS, our Lagrange relaxation ILPF-for-RPS can decrease the running time by more than 90% in most cases.**

***Index Terms*—IEEE 802.16 mesh network, Lagrange relaxation, Linear programming, NP-complete, WiMAX technology.**

I. Introduction

The IEEE 802.16 standard, also known as the WiMAX technology, is a technology of wireless network access which can service a large area. Its transmission rate exceeds 100 Mbps [1], which qualifies itself to fit the category of high-speed wireless broadband network technology. Since the IEEE 802.16 standard adopts the multi-hop technique to deal with data packets among subscriber stations (SSs), only a few base stations (BSs) are required to cover a large metropolitan area. For this reason, an efficient routing and packet scheduling (RPS) algorithm is necessary to deal with SS-to-SS and SS-to-BS data transmissions. In fact, the RPS problem has recently become an important research topic in the multi-hop IEEE 802.16 standard [4][9][10][11][12][15]. In this paper, we study the RPS problem in the IEEE 802.16 network and propose a fast method to solve it.

The multi-hop IEEE 802.16 standard can be subdivided into two types: mesh networks and mobile multi-hop relay networks. In this paper, mesh networks are concerned. In a mesh network, the objective of the RPS problem is to maximize the throughput of the network. In other words, the RPS problem is to design a fast scheduling scheme to meet the requirement of all the SSs so that all of the packets can be delivered to the BS while minimizing the number of time slots and prohibiting the interference between any two packets.

Integer or mixed integer linear programming formulations [8] have been adopted by many researchers to solve various problems in wireless networks [2][3][5][14]. Similarly, there has existed an integer linear programming formulation (ILPF) for the RPS problem, named as ILPF-for-RPS, which can find its optimal solution [9]. However, the execution time of MIPF-for-MSC is very long when the number of SSs is large. This is because the authors of literature [9] have proven that the RPS problem is NP-complete for a general network topology. This means that it must take exponential time to find the optimal solution of the RPS problem.

As an approach to the ILPF of a NP-complete problem, an efficient computational methodology was proposed around 1970, namely, the Lagrange relaxation method [6][7]. The basic idea is that some constraints of a given ILPF can be relaxed so as to reduce the solution space, which in turn shortens the solution-searching time. In other words, this method relaxes the constraints which may otherwise make the running time of the ILPF of a combinatorial optimization problem become exponential. These relaxed constraints are merged into the objective function such that the original ILPF becomes a Lagrange relaxation ILPF. In general, an optimal solution to the resultant Lagrange relaxation ILPF can be obtained within a shorter period of time.

In this paper, the Lagrange relaxation method is applied to shrink the solution space of the ILPF-for-RPS for the purpose of shortening the solution-searching time. First, we theoretically transform the ILPF-for-RPS into a Lagrange relaxation ILPF-for-RPS, whose objective function is then proved to be a concave function. This Lagrange relaxation ILPF-for-RPS simplifies the original ILPF-for-RPS and can be used to find an optimal solution for the RPS problem within a shorter time. In fact, to speed up the search of an optimal solution, we have cut down the solution space of the original ILPF-for-RPS. Our computer simulations show a decrease of the solution space by 37.73%. In other words, if the solution space must be searched by the original ILPF-for-RPS [9] in is 100%, our Lagrange relaxation ILPF-for-RPS can find an optimal time slot schedule with 62.27% of its solution space. Our method is feasible. This is because different schedules of minimum time slots usually exist in the RPS problem. We eliminate 37.73% of the solution space without completely eliminating all the minimum time slot schedules.

According to the computer simulations, in contrast to the original ILPF-for-RPS in [9], our Lagrange relaxation ILPF-for-RPS can attain a minimum time slot schedule within a shorter time for the RPS problem. To be more specific, compared with the original ILPF-for-RPS, our Lagrange relaxation ILPF-for-RPS can decrease the running time by more than 90% in most cases. In conclusion, our Lagrange relaxation method is demonstrated to be valuable for its contribution of a shorter solution time of the RPS problem.

The rest of this paper is organized as follows: In Section II, the RPS problem is described in detail and defined formally. In addition, an example is given to illustrate the RPS problem and its main constraints for packet transmissions. In Section III, the known ILPF-for-RPS is presented. In Section

IV, we apply the Lagrange relaxation method to the known ILPF-for-RPS, theoretically transform it into a Lagrange relaxation ILPF-for-RPS, and then prove the objective function of the Lagrange relaxation ILPF-for-RPS to be a concave function. In Section V, the performance of our Lagrange relaxation ILPF-for-RPS is evaluated and compared with that of the original ILPF-for-RPS through computer simulations. Finally, in Section VI, the conclusions of this study are drawn and our main contributions are stated.

II. Problem Description

In this section, we present the RPS problem. In the RPS problem, a centralized scheduling is adopted. Therefore, the BS serves as the centralized schedulers for the entire network. In the following, the network considered is assumed to contain only one BS and several SSs. Each SS has packets to send to the BS. In this paper, we assume that the routing must follows the three constraints proposed by the literature [9].

(1) A SS cannot send and receive simultaneously.
(2) There must be only one transmitter in the neighborhood of a receiver.
(3) There must be only one receiver in the neighborhood of a transmitter.

A. Problem statement

Given a graph $G=(V,E,w)$ and a transmission schedule $S=(s_1,s_2,\cdots,s_m)$, where set V consists of a single BS: v_0 and multiple SSs: $v_1,v_2,\cdots,v_n$. E is the set of all links in G. If v_i and v_j are within transmission ranges of each other, then there exists a link $(v_i,v_j)\in E$. A packet-transmission function $w:V\rightarrow R^+$ is defined. For each SS, it gives the number of packets that will be sent to BS. $s_t=\{V_t,E_t\}$ is the transmission schedule at timeslot t, where V_t is the transmitter set and E_t is the link set at timeslot t. The RPS problem is to find a routing tree and a transmission schedule set S, so that only a minimal amount of timeslot is used to send all packets from each SS to BS.

B. An example to illustrate the RPS problem

Let us use the example in Figure 1 to illustrate the RPS problem. The network in Figure 1 consists of a BS: v_0 and five SSs: v_1, v_2, v_3, v_4 and v_5. In Figure 1, the number within braces adjacent to each node v_i shows the number of packets to be sent from v_i to v_0. For example, the number '1' in the braces next to v_1 means that there is one packet needs to be sent from v_1 to v_0 for further process. For simplicity, in this example, we assume that each of the five SSs: v_1,v_2,v_3,v_4 and v_5 has only one single packet to be sent to BS. That is, there are five packets in total to be sent to v_0. Each column of the table in Figure 1 corresponds to each node: v_1, v_2, v_3, v_4 and v_5. Each row represents a serial number denoting the timeslot. There are five timeslots in total which means all the packets from all the SSs can be sent to BS within five timeslots.

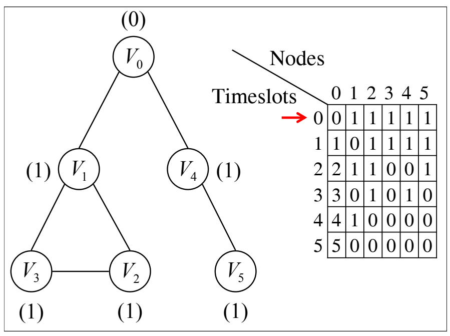

Fig. 1 An example to illustrate the RPS problem.

In the beginning, the first row represents the state at timeslot 0 which no packets is received by v_0 and each SS has one packet to send. Next, at timeslot 1, SS v_1 sent a packet to BS v_0. Hence, the value with v_1 is decreased to 0. At the same time, the value with v_0 is increased by 1 to means that a packet is already received from v_1. Meanwhile (at timeslot 1), all the nodes must obey the three routing constrains. As a result, v_1 can not receive packets from v_2 or v_3 and send packets to v_0. Similarly, v_0 can not receive packets from v_4 and receive packets from v_1. Therefore, there are still four packets waiting to be sent after v_1 sent a packet to v_0. At timeslot 2 v_1 served as a repeater while v_3 sent a packet, i.e., v_1 retransmits the packet v_0. At the same time, v_4 also sends a packet to v_0. Therefore, there remain three packets to be sent in the entire network. At timeslot 3, v_1 sends the packet coming from v_3 to v_0, and v_5 sends its packet to v_4. Hence, there are two packets waiting to be sent. At timeslot 4, v_2 then send its packet to v_1 while v_4 sends packet to v_0. Thus, only one packet is left to be transmitted. Finally, at timeslot 5, v_1 sends the packet to v_0. Thus, all the packets in the entire network have arrived at BS. Therefore, at least five timeslots are needed to send all the five packets from SSs to BS in Figure 1.

III. A KNOWN ILPF FOR THE RPS PROBLEM

In this section, we described a known ILPF for RPS problem: ILPF-for-RPS, provided by [9].

Variables used in ILPF-for-RPS are defined as follows: Y_t is a Boolean variable at timeslot t. When all the packets have arrived at BS, $Y_t = 1$; Otherwise, $Y_t = 0$. R_{ij} is a Boolean variable. When v_j is father of v_i, $R_{ij} = 1$; Otherwise, $R_{ij} = 0$. X_{ijt} is a Boolean variable. When v_i sends packets to v_j at timeslot t, $X_{ijt} = 1$; Otherwise, $X_{ijt} = 0$. w_{it} denotes the number of packets to be sent in node v_i at timeslot t. A_t denotes the number of packets that haven't arrived at BS. U is an upper bound of timeslots in need. It is not hard to see that the number of required timeslots reaches the highest value when only one node sends a packet at each timeslot. That is, the highest value is equal to the sum of the products of packets in each node and the least hops from the node to the root.

Based on the above notation and definition, ILPF-for-RPS can be described as follows:

Objective function:

$$Z_{\mathrm{IP(RPS)}} = \text{Minimize} \sum_{t=1}^{U} tY_t \tag{1}$$

Subject to constrains:

$$R_{ij} \le E_{ij} \quad \forall i \in \{1,\cdots,n\}, \forall j \in \{0,\cdots,n\} \tag{2}$$

$$\sum_{j=0}^{n} R_{ij} = 1 \quad \forall i \in \{1,\cdots,n\} \tag{3}$$

$$X_{ijt} \le R_{ij} \quad \forall i \in \{1,\cdots,n\}, \forall j \in \{0,\cdots,n\}, \forall t \in \{1,\cdots,U\} \tag{4}$$

$$\begin{cases} X_{ikt} + X_{jkt} \le 1 \quad \forall i \ne j & (5.1) \\ X_{ijt} + X_{jkt} \le 1 & (5.2) \end{cases}$$
$$\forall i, j \in \{1,\cdots,n\},\ k \in \{0,\cdots,n\}, t \in \{1,\cdots,U\}$$

$$w_{jt} = w_{j(t-1)} + \sum_i X_{ijt} - \sum_k X_{jkt} \quad \forall t \in \{1,\cdots,U\} \tag{6}$$

$$A_t = \sum_{i=1}^{n} w_{it} \quad \forall t \in \{1,\cdots,U\} \tag{7}$$

$$\sum_{t=1}^{U} Y_t = 1 \quad (8)$$

$$A_t \le A_0(1 - Y_t) \quad \forall t \in \{1, \cdots, U\} \quad (9)$$

$$Y_t = \begin{cases} 1, & A_t = 0 \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

$$R_{ij} = \begin{cases} 1, & v_j \text{ is the parent of } v_i \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

$$X_{ijt} = \begin{cases} 1, & v_i \text{ sends a packet to } v_j \text{ at timeslot } t \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

Inequality (2) means that $R_{ij} = 1$ only when link (v_i, v_j) exists. Equation (3) shows that each v_i only has a single parent node. Inequality (4) requires that v_i may send packets to v_j only when link (v_i, v_j) exists in a multicast tree. According to the three routing constrains proposed by [9], Inequalities (5.1) and (5.2) are obtained. Equation (6) requires that the number of packets queued in v_i at timeslot t must be equal to the number of the packets remained from the previous timeslots plus the number of packets received at the present timeslot, and minus the number of packets just sent. Equation (7) accumulates all the packets that have not arrived at BS in Variable A_t. Equation (8) and Inequality (9) turn Y_t into 1 when all the packets are received by BS at timeslot t. The ranges of Y_t, R_{ij}, and X_{ijt} are specified by functions (10) to (12), respectively.

The computational complexity of ILPF-for_RPS is $O(n^2U)$, where n is the number of nodes and U is the unknown upper bound of total timeslots. Therefore, constrains (8) and (9) of the ILPF-for-RPS will increase along with, such that the time for solving RPS problem will grow accordingly. In order to shorten the time for finding a solution, we will relax constrains (8) and (9).

IV. Using the Lagrange Relaxation Method to Analyze ILPF-for-RPS

A. Introduction to the Lagrange relaxation method

As an approach to the ILPF of a NP-complete problem, an efficient computational methodology was proposed around 1970, namely, the Lagrange relaxation method. The basic idea is that some constraints of a given ILPF can be relaxed so as to reduce the solution space, which in turn shortens the solution-searching time. In other words, this method relaxes the constraints which may otherwise make the running time of the ILPF of a combinatorial optimization problem become exponential. These relaxed constraints are merged into the objective function such that the original ILPF becomes a Lagrange relaxation ILPF. In general, an optimal solution to the resultant Lagrange relaxation ILPF can be obtained within a shorter period of time.

Let us use the following ILPF to illustrate the Lagrange relaxation method.

Objective function:

$$Z_{\mathrm{IP_a}} = \text{Minimize} \sum_{t=1}^{n} c_i x_i \quad (13)$$

Constrains:

$$\sum_{j=1}^{m} \sum_{i=1}^{n} a_{ij} x_i \le b_j \quad (14)$$

$x_i \geq 0$ and $x_i \in \mathrm{N} \quad \forall i \in \{1, \cdots, n\}$ (15)

Constrain (15) denotes all variables $x_1, x_2, \cdots, x_k$ fall in the range of natural number N, and $\{x_i \mid i = 1, 2...k\} \geq 0$. In constrain (14), the sum of the products of each x_i and a_{ij} is bounded by b_j. Now, suppose that the constrain (14) will make the objective function be unable to minimize $\sum_{i=1}^{n} c_i x_i$ under polynomial time, we then relax constrain (14) and merge it into the objective function to form a Lagrangian relaxation ILPF. Thus, the Lagrange relaxation ILPF can easily find the minimum of $\sum_{i=1}^{n} c_i x_i$. The Lagrange relaxation ILPF is composed of the objective function (13) and Constrains (14) and (15).

Relaxed objective function:

$$\mathrm{Z}_{\mathrm{LR_a}} = \text{Minimize} \sum_{i=1}^{n} c_i x_i + \lambda \cdot \left(\sum_{j=1}^{m} \sum_{i=1}^{n} a_{ij} x_i - b_j\right) \tag{16}$$

Relaxed constrain:

$$x_i \geq 0 \text{ and } x_i \in \mathrm{N} \quad \forall i \in \{1, \cdots, n\}, \tag{17}$$

where λ is Lagrange multiplier. We can modify λ so that the solution space can be reduced, which in turn speed up the solution-searching process.

B. Definitions and theorems

In this subsection, we propose several Theorems for RPS problem. First of all, we relax Equation (8) (i.e. $\sum_{t=1}^{U} Y_t = 1$) and Inequality (9) (i.e. $A_t \leq A_0(1 - Y_t) \quad \forall t$) and merge them into objective function (1) (i.e. $\mathrm{Z}_{\mathrm{IP(RPS)}} = \text{Minimize} \sum_{t=1}^{U} t Y_t$) to transform MILP-for-RPS into a Lagrange relaxation MILP-for-RPS. This Lagrange relaxation MILP-for-RPS can shrink the solution space of original one and find an optimal solution in short time.

The Lagrange relaxation MILP-for-RPS is defined as follows:

Relaxed objective function:

$$\mathrm{Z}_{\mathrm{LR(RPS)}}(\lambda_1, \lambda_2) = \text{Minimize} \left\{ \sum_{t=1}^{U} t Y_t + \lambda_1 \cdot \left(\sum_{t=1}^{U} Y_t - 1\right) + \lambda_2 \cdot \sum_{t=1}^{U} \left[A_t - A_0(1 - Y_t)\right] \right\} \tag{18}$$

where λ_1 and λ_2 are Lagrange multipliers.

Subject to Relaxed constrains:

Constrains (2), (3), (4), (5), (6), (7), (10), (11), and (12).

Definition 1: If there is a function $Z(x): R \to R$, and there exists three points x_1, x_2, and x_3 in [a, b], such that $a < x_1 < x_2 < x_3 < b$ and $Z(x_2) \geq L(x_2)$, where $L(x)$ is a linear equation through Points $(x_1, Z(x_1))$ and $(x_3, Z(x_3))$, then $Z(x)$ is a concave function.

Theorem 1: Let $Q = \{Y_t^k \mid k = 1, 2, \cdots, K\}$ be a feasible solution set, where $Q \in \mathrm{N}$. If Q is a finite set, i.e., $Q < \infty$, then

$$\mathbf{Z}_{\mathbf{LR(RPS)}}(\lambda_1, \lambda_2) = \textbf{Minimize} \left\{ \sum_{t=1}^{U} t Y_t^k + \lambda_1 \cdot \left(\sum_{t=1}^{U} Y_t^k - 1 \right) + \lambda_2 \cdot \sum_{t=1}^{U} \left[A_t - A_0\left(1 - Y_t^k\right) \right] \middle| \, Y_t^k \in Q \right\}$$

is a concave function.

Proof: Since $Q=\{Y_t^k \mid k=1,2,\cdots,K\}$ is a finite feasible solution set, $Y_t^1,Y_t^2,\cdots,Y_t^K$ are feasible solutions. Therefore, Function $Z_{LR(RPS)}(\lambda_1,\lambda_2)$ can be expressed as follows:

$$\mathbf{Z}_{\mathbf{LR(RPS)}}(\lambda_1,\lambda_2)=\underset{1\le k\le K}{\mathbf{Minimize}}\left\{\sum_{t=1}^{U} tY_t^k+\lambda_1\cdot\left(\sum_{t=1}^{U}Y_t^k-1\right)+\lambda_2\cdot\sum_{t=1}^{U}\left[A_t-A_0\left(1-Y_t^k\right)\right]\right\}$$

If $Z_{LR(RPS)}(\lambda_1,\lambda_2)$ has the minimum feasible solution Y_t^q, then

$$\mathbf{Z}_{\mathbf{LR(RPS)}}(\lambda_1,\lambda_2)=\sum_{t=1}^{U} tY_t^q+\lambda_1\cdot\left(\sum_{t=1}^{U}Y_t^q-1\right)+\lambda_2\cdot\sum_{t=1}^{U}\left[A_t-A_0\left(1-Y_t^q\right)\right].$$

Because $\sum_{t=1}^{U}Y_t-1=0$ by (8), we can temporally ignore λ_1 so that we rewrite $Z_{LR(RPS)}(\lambda_1,\lambda_2)$ as

$$\mathbf{Z}_{\mathbf{LR(RPS)}}(\lambda_2)=\sum_{t=1}^{U} tY_t^q+\lambda_2\cdot\sum_{t=1}^{U}\left[A_t-A_0\left(1-Y_t^q\right)\right].$$

Now suppose that there exists a Lagrange multiplier $\lambda_2^*=\alpha\cdot\lambda_2^a+(1-\alpha)\cdot\lambda_2^b$ such that $Z_{LR(RPS)}(\lambda_2^*)$ has the largest solution space under the minimum feasible solution Y_t^q, where $\alpha=[0,1]$. We have

$$\begin{aligned}
&\mathbf{Z}_{\mathbf{LR(RPS)}}(\lambda_2^*)=\mathbf{Z}_{\mathbf{LR(RPS)}}\left(\alpha\cdot\lambda_2^a+(1-\alpha)\cdot\lambda_2^b\right)\\
&=\sum_{t=1}^{U} tY_t^q+\left(\alpha\cdot\lambda_2^a+(1-\alpha)\cdot\lambda_2^b\right)\cdot\left(\sum_{t=1}^{U}\left[A_t-A_0\left(1-Y_t^q\right)\right]\right)\\
&\ge\alpha\cdot\mathbf{Z}_{\mathbf{LR(RPS)}}(\lambda_2^a)+(1-\alpha)\cdot\mathbf{Z}_{\mathbf{LR(RPS)}}(\lambda_2^b)\\
&=\alpha\cdot\left(\underset{1\le k\le K}{\mathbf{Minimize}}\left\{\sum_{t=1}^{U} tY_t^k+\lambda_2^a\cdot\sum_{t=1}^{U}\left[A_t-A_0\left(1-Y_t^k\right)\right]\right\}\right)\\
&+(1-\alpha)\cdot\left(\underset{1\le k\le K}{\mathbf{Minimize}}\left\{\sum_{t=1}^{U} tY_t^k+\lambda_2^b\cdot\sum_{t=1}^{U}\left[A_t-A_0\left(1-Y_t^k\right)\right]\right\}\right)
\end{aligned}$$

From the $Z_{LR(RPS)}$ diagram in Figure 2, we know that the corresponding values on line u or curve v can be obtain when parameter λ_2^* is given. Because parameter λ_2^* is between λ_2^a and λ_2^b, the corresponding value $Z_{LR(RPS)}(\lambda_2^*)$ on line $Z_{LR(RPS)}$ when parameter λ_2^* is substituted into line u is the linear combination of $Z_{LR(RPS)}(\lambda_2^a)$ and $Z_{LR(RPS)}(\lambda_2^b)$. Hence,

$$Z_{LR(RPS)}(\lambda_2^*)=\alpha\cdot Z_{LR(RPS)}(\lambda_2^a)+(1-\alpha)\cdot Z_{LR(RPS)}(\lambda_2^b).$$

Because we assume that with $\lambda_2^*=\alpha\cdot\lambda_2^a+(1-\alpha)\cdot\lambda_2^b$, $Z_{LR(RPS)}(\lambda_2^*)$ have the largest solution space under the minimum feasible solution Y_t^q, and according to the inference above, the value $Z_{LR(RPS)}(\lambda_2^*)$ on line $Z_{LR(RPS)}$ corresponding to λ_2^* is greater than the medium value. This result is only satisfied at the corresponding value on curve v. It implies that $Z_{LR(RPS)}(\lambda_2)$ has the feature of concave function. These inferences also satisfy the sufficient and necessary conditions for Definition 1.

We know that substituting some λ_2 into $Z_{LR(RPS)}(\lambda_2)$ makes it has the largest solution space under the minimum feasible solution Y_t^q. This implies that there is only one λ_2 making $Z_{LR(RPS)}(\lambda_2)$ possesses the largest solution. Therefore, we suppose that $\lambda_2=\lambda_2^*=\alpha\cdot\lambda_2^a+(1-\alpha)\cdot\lambda_2^b$ makes $\mathbf{Z}_{\mathbf{LR(RPS)}}(\lambda_2^*)$ has the largest solution space under the minimum feasible solution Y_t^q.

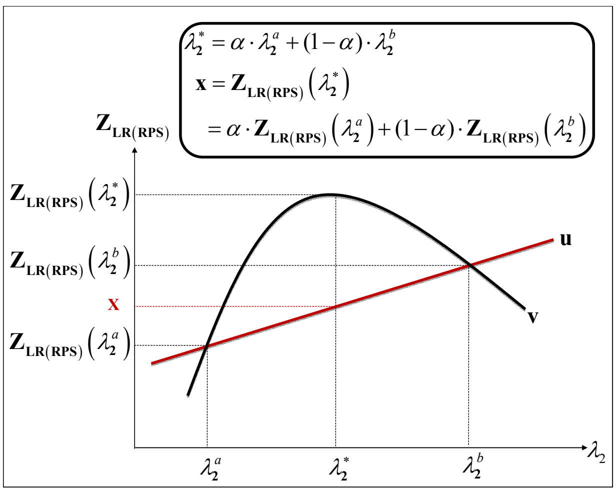

Fig. 2 Diagram of $\mathbf{Z}_{\mathbf{LR(RPS)}}$ in 2-dimension.

It means that the solution space of $\mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2^*\right)$ is greater than those of both $\mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2^a\right)$ and $\mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2^b\right)$. That is to say, $\mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2^*\right) > \mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2^a\right)$ and $\mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2^*\right) > \mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2^b\right)$. In this case, we rationalized $\mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2^*\right) \geq \alpha \cdot \mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2^a\right) + (1-\alpha)\cdot \mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2^b\right)$, so it can be claimed that $\mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2\right)$ is a concave function. Hence, we have showed that $\mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2\right) = \mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_1, \lambda_2\right)$ is a concave function.□

Theorem 2: If function $\mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2\right) = \mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_1, \lambda_2\right)$ is a concave function, then there exists a parameter λ_2^* and a set $S = \left\{s^i \mid i = 1, 2, \cdots, n\right\}$ such that $\mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2^*\right) + s^i \cdot \left(\lambda_2 - \lambda_2^*\right) \geq \mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2\right)$ holds, where $\forall \lambda, \lambda^*, s^1, s^2, \cdots, s^n \in R$.

Proof: First of all, suppose that $\mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2\right)$ is a concave function. A set $I = \left\{(\lambda_2, Z) \mid Z \leq \mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2\right)\right\}$ is given such that (λ_2^a, Z^a) and (λ_2^b, Z^b) both belong to I. Thus, we can obtain a point (λ_2^*, Z^*) between (λ_2^a, Z^a) and (λ_2^b, Z^b) such that

$$
\begin{aligned}
(\lambda_2^*, Z^*) &= \alpha \cdot (\lambda_2^a, Z^a) + (1-\alpha) \cdot (\lambda_2^b, Z^b) \\
&= (\alpha \cdot \lambda_2^a + (1-\alpha) \cdot \lambda_2^b, \quad \alpha \cdot Z^a + (1-\alpha) \cdot Z^b)
\end{aligned}
$$

where $\alpha = [0, 1]$. By Theorem 1, we know that substituting $\alpha \cdot \lambda_2^a + (1-\alpha) \cdot \lambda_2^b$ into the concave function $\mathbf{Z}_{\mathbf{LR(RPS)}}(\cdot)$ can satisfy the following condition:

$$
\begin{aligned}
&\mathbf{Z}_{\mathbf{LR(RPS)}}\left(\alpha \cdot \lambda_2^a + (1-\alpha) \cdot \lambda_2^b\right) \\
&\geq \alpha \cdot \mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2^a\right) + (1-\alpha) \cdot \mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2^b\right) \\
&= \alpha \cdot Z^a + (1-\alpha) \cdot Z^b.
\end{aligned}
$$

By Theorem 1, it can be found that the concave function $\mathbf{Z}_{\mathbf{LR(RPS)}}\left(\alpha \cdot \lambda_2^a + (1-\alpha) \cdot \lambda_2^b\right)$ will be always greater than or equal to $\alpha \cdot Z^a + (1-\alpha) \cdot Z^b$. Therefore, it is sure that the function passes through (λ_2^a, Z^a) and (λ_2^b, Z^b) is not linear. In this case, it can be guaranteed that the point $\left(\alpha \cdot \lambda_2^a + (1-\alpha) \cdot \lambda_2^b, \alpha \cdot Z^a + (1-\alpha) \cdot Z^b\right)$ between (λ_2^a, Z^a) and (λ_2^b, Z^b) belongs to set I. So, it can be claimed that I is a convex set.

Suppose that there exists a point $\left(\lambda_2^*, \mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2^*\right)\right)$ located at the margin of the solution space of set I, and there exists an orthogonal tangent plane $\mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2^*\right) + s^i \cdot \left(\lambda_2 - \lambda_2^*\right)$ which passes through $\left(\lambda_2^*, \mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2^*\right)\right)$ and is generated by s^i. Let us let $\lambda_2 < \lambda_2^*$, then the following can be satisfied:

$$
\mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2^*\right) + s^i \cdot \left(\lambda_2 - \lambda_2^*\right) > \mathbf{Z}_{\mathbf{LR}}\left(\lambda_2\right)
$$

where $\lambda_2^* = \alpha \cdot \lambda_2^a + (1-\alpha) \cdot \lambda_2^b$ and $\alpha = [0, 1]$.

With the above procedure, $\mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2^*\right) + s^i \cdot \left(\lambda_2^a - \lambda_2^*\right) > \mathbf{Z}_{\mathbf{LR}}\left(\lambda_2^a\right)$ and $\mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2^*\right) + s^i \cdot \left(\lambda_2^b - \lambda_2^*\right) > \mathbf{Z}_{\mathbf{LR}}\left(\lambda_2^b\right)$ can be both fulfilled. By further expanding these conditions, we can obtain the following inequality:

$$
\begin{aligned}
&\mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2^*\right) + s^i \cdot \left\{\left[\alpha \cdot \lambda_2^a + (1-\alpha) \cdot \lambda_2^b\right] - \lambda_2^*\right\} \\
&\geq \alpha \cdot \mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2^a\right) + (1-\alpha) \cdot \mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2^b\right)
\end{aligned}.
$$

Since parameter $s^i \cdot \left\{\left[\alpha \cdot \lambda_2^a + (1-\alpha) \cdot \lambda_2^b\right] - \lambda_2^*\right\}$ is a very small and negative value, it is even eliminated and $\mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2^*\right) \geq \alpha \cdot \mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2^a\right) + (1-\alpha) \cdot \mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2^b\right)$ still holds. □

Definition 2: If $\mathbf{Z}_{\mathbf{LR(RPS)}} : R \to R$ is a concave function, and there exist multiplier $\lambda_2^* \in R$ and parameter $s \in R$ such that $\mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2^*\right) + s \cdot \left(\lambda_2 - \lambda_2^*\right) \geq \mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2\right) \quad \forall \lambda_2 \in R$, then we denote s as a subgradient of $\mathbf{Z}_{\mathbf{LR(RPS)}}(\cdot)$ at λ_2^*, and the set consisting of subgradients generated by $\mathbf{Z}_{\mathbf{LR(RPS)}}(\cdot)$ at λ_2^* is denoted as $\frac{\partial}{\partial \lambda_2^*} \mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2^*\right)$.

Theorem 3: Suppose linear programming relaxation formulation $\mathbf{Z}_{\mathbf{LR(RPS)}} : R \to R$ is a concave function such that $\mathbf{Z}_{\mathbf{LR(RPS)}}(\cdot)$ has the largest feasible solution space (i.e., $\max\{\mathbf{Z}_{\mathbf{LR(RPS)}}(\lambda_2) \mid \lambda_2 \in R, \lambda_2 \geq 0\}$), after Lagrange multiplier λ_2 converges to λ_2^* and is substitute into $\mathbf{Z}_{\mathbf{LR(RPS)}}(\cdot)$. Hence, λ_2 converges to λ_2^* if and only if

$$s^i = \frac{\partial}{\partial \lambda_2^*} \mathbf{Z}_{\mathbf{LR(RPS)}}(\lambda_2^*) = 0 .$$

Proof: By Definition 2, $\mathbf{Z}_{\mathbf{LR(RPS)}}(\lambda_2^*) + s^i \cdot (\lambda_2 - \lambda_2^*) \geq \mathbf{Z}_{\mathbf{LR(RPS)}}(\lambda_2)$ holds and implies $s^i \cdot (\lambda_2 - \lambda_2^*) \geq \mathbf{Z}_{\mathbf{LR(RPS)}}(\lambda_2) - \mathbf{Z}_{\mathbf{LR(RPS)}}(\lambda_2^*)$. Therefore, suppose that the necessary and sufficient condition of λ_2 converging to λ_2^* is a subgradient $s^i = \partial / \partial \lambda_2^* \mathbf{Z}_{\mathbf{LR(RPS)}}(\lambda_2^*) = 0$, then it is clearly that $0 \cdot (\lambda_2 - \lambda_2^*) \geq \mathbf{Z}_{\mathbf{LR(RPS)}}(\lambda_2) - \mathbf{Z}_{\mathbf{LR(RPS)}}(\lambda_2^*)$ holds, which leads $\mathbf{Z}_{\mathbf{LR(RPS)}}(\lambda_2^*) \geq \mathbf{Z}_{\mathbf{LR(RPS)}}(\lambda_2)$. Because the feasible solution space of $\mathbf{Z}_{\mathbf{LR(RPS)}}(\lambda_2^*)$ is close to the one of linear programming formulation $\mathbf{Z}_{\mathbf{IP(RPS)}}$ when Lagrange multiplier λ_2 converges to λ_2^*, the feasible solution space of $\mathbf{Z}_{\mathbf{LR(RPS)}}(\lambda_2^*)$ is larger than that of $\mathbf{Z}_{\mathbf{LR(RPS)}}(\cdot)$ at any other $\lambda_2 \in R$, which justifies our assumption. □

Theorem 4: If there exist several feasible solutions $Y_t^1, Y_t^2, \cdots, Y_t^K$ for $\mathbf{Z}_{\mathbf{LR(RPS)}}(\cdot)$, in which a feasible solution Y_t^k makes the following be held:

$$\mathbf{Z}_{\mathbf{LR(RPS)}}(\lambda_2^*) = \underset{k \in K}{\mathbf{Minimize}} \left\{ \sum_{t=1}^{U} t Y_t^k + \lambda_2^* \cdot \sum_{t=1}^{U} \left[A_t - A_0 \left(1 - Y_t^k\right) \right] \right\}$$

Therefore, given a set

$$M = \left\{ i \middle| \mathbf{Z}_{\mathbf{LR(RPS)}}(\lambda_2^*) = \sum_{t=1}^{U} t Y_t^i + \lambda_2^* \cdot \sum_{t=1}^{U} \left[A_t - A_0 \left(1 - Y_t^i\right) \right] \right\},$$

then there exists a subgradient for any $i \in M$

$$s^i = \frac{\partial}{\partial \lambda_2^*} \mathbf{Z}_{\mathbf{LR(RPS)}}(\lambda_2^*) = \sum_{t=1}^{U} \left[A_t - A_0 \left(1 - Y_t^i\right) \right]$$

where s^i is a subgradient of $\mathbf{Z}_{\mathbf{LR(RPS)}}(\cdot)$ at λ_2^*, and $s^i \subseteq \partial / \partial \lambda_2^* \mathbf{Z}_{\mathbf{LR(RPS)}}(\lambda_2^*)$. If each element i in set M satisfies the following equation

$$s^i = \frac{\partial}{\partial \lambda_2^*} \mathbf{Z}_{\mathbf{LR(RPS)}}(\lambda_2^*) = \sum_{t=1}^{U} \left[A_t - A_0 \left(1 - Y_t^i\right) \right],$$

then at λ_2^*, $\mathbf{Z}_{\mathbf{LR(RPS)}}(\cdot)$ has multiple subgradients:

$$s^i \subseteq \frac{\partial}{\partial \lambda_2^*} \mathbf{Z}_{\mathbf{LR(RPS)}}(\lambda_2^*), i = 1, 2, \cdots, n .$$

Proof:

$$\begin{aligned}
& s^i \cdot (\lambda_2 - \lambda_2^*) \\
& = \sum_{t=1}^{U} \left[A_t - A_0 \left(1 - Y_t^i\right) \right] \cdot (\lambda_2 - \lambda_2^*) \\
& = \lambda_2 \cdot \left\{ \sum_{t=1}^{U} \left[A_t - A_0 \left(1 - Y_t^i\right) \right] \right\} - \lambda_2^* \cdot \left\{ \sum_{t=1}^{U} \left[A_t - A_0 \left(1 - Y_t^i\right) \right] \right\} \\
& \geq \mathbf{Z}_{\mathbf{LR(RPS)}}(\lambda_2) - \mathbf{Z}_{\mathbf{LR(RPS)}}(\lambda_2^*) \\
& \Rightarrow \mathbf{Z}_{\mathbf{LR(RPS)}}(\lambda_2^*) + s^i \cdot (\lambda_2 - \lambda_2^*) \geq \mathbf{Z}_{\mathbf{LR(RPS)}}(\lambda_2)
\end{aligned}$$

By Definition 2, we know that s^i is a subgradient (s^i is defined by $s^i = \frac{\partial}{\partial \lambda_2^*} \mathbf{Z}_{\mathbf{LR(RPS)}}(\lambda_2^*) = \sum_{t=1}^{U} \left[A_t - A_0 \left(1 - Y_t^i\right) \right]$).

With a similar inference, it can be justified that each element $i = 1, 2, \cdots, n$ in set M satisfies the equation

$$s^i = \frac{\partial}{\partial \lambda_2^*} \mathbf{Z}_{\mathbf{LR(RPS)}}(\lambda_2^*) = \sum_{t=1}^{U} \left[A_t - A_0 \left(1 - Y_t^i\right) \right].$$

Hence at λ_2^*, $\mathbf{Z}_{\mathbf{LR(RPS)}}(\cdot)$ has multiple subgradients:

$$s^i \subseteq \frac{\partial}{\partial \lambda_2^*} \mathbf{Z}_{\mathbf{LR(RPS)}}(\lambda_2^*), i = 1, 2, \cdots, n$$ □

C. The choice of Lagrangian parameter λ_2

By Theorem 1 and Theorem 2, $\mathbf{Z}_{\mathbf{LR(RPS)}}(\lambda_2)$ has been proved to be a concave function. Therefore, there must exist exactly one λ_2^* in our Lagrangian ILP-for-RPS such that the solution space of $\mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2^*\right)$ contains the optimal feasible solution Y_t^*. According to Theorem 4, we first substitute λ_2^* into $\mathbf{Z}_{\mathbf{LR(RPS)}}()$, and then we choose a subgradient s^* from the generated subgradient set $\partial/\partial\lambda_2^*\mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2^*\right)$. If s^* complies with Theorem 3 (i.e., $s^i=\partial/\partial\lambda_2^*\mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda_2^*\right)=0$), then λ_2 has converged to the optimal λ_2^*. Finally, according to those theorems above, we establish Algorithm 1 (see Figure 3) to generate Lagrangian parameter λ_2.

V. COMPUTER SIMULATIONS

In the computer simulation, our hardware contains a PC with two Intel(R) Pentium(R) IV CPU runs at 3.40GHz and 1014MB RAM. In software aspect, we use C/C++ and LINGO 8.0 [13].

Figure 4 shows our computer simulation results. When the Lagrange parameter λ_2 comes to 0.061756, the resultant solution space is 62.27% of the one searched by the original ILPF-for-RPS. In other words, we reduce 37.73% computation for scheduling, which in turn saves about 1/3 computational time.

The performance benchmarks reflect CPU execution time. In Table 1, with 9 nodes, the original ILPF-for-RPS $Z_{IP(RPS)}$ needs 7 minutes and 50 seconds to solve the RPS problem while our Lagrange relaxation ILPF-for-RPS $Z_{LR(RPS)}$ only spends 10 seconds to find the optimal solution. In Table 1, there is no case with more than 17 nodes because the execution time of $Z_{IP(RPS)}$ exceeds 3600 minutes. The execution time is too long, so we stop to increase the number of nodes in our simulations. Compared with the original ILPF-for-RPS, Table 1 shows that with different numbers of nodes: 9, 11, 13, and 15, our Lagrange relaxation ILPF-for-RPS can decrease running time by 97.87%, 90.24%, 69.91%, and 96.50%, respectively. While a network containing 17 nodes, the CPU execution time can be reduced from more than 3600 minutes to 31 minutes and 36 seconds.

Algorithm 1: The generation of Lagrangian parameter λ_2

Step 1: Randomly select a feasible solution $\mathbf{Z}_{\mathbf{upper}}(t)$ from $\mathbf{Z}_{\mathbf{IP(RPS)}}$, and initialize the Lagrangian parameter $\lambda^{\mathbf{t=1}}=0$ and an adjusting parameter $\mu_{\mathbf{t=1}}=2$

Step 2: Substitute current $\lambda^{\mathbf{t}}$ into $\mathbf{Z}_{\mathbf{LR(RPS)}}$ to seek the solution to $\mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda^{\mathbf{t}}\right)$

Step 3: Use x_i to calculate a subgradient

$$\mathbf{s}^{\mathbf{t}}=\frac{\partial}{\partial\lambda}\mathbf{Z}_{\mathbf{LR(RPS)}}$$

Step 4: Use the parameters above to estimate

$$\text{step-size } \theta_{\mathbf{t}}=\frac{\left(\mathbf{Z}_{\mathbf{Upper}}-\mathbf{Z}_{\mathbf{LR(RPS)}}\left(\lambda^{\mathbf{t}}\right)\right)}{\left\|\mathbf{s}^{\mathbf{t}}\right\|^2}\times\mu_{\mathbf{t}}$$

Step 5: Use subgradient $\mathbf{s}^{\mathbf{t}}$ and step-size $\theta_{\mathbf{t}}$ to adjust Lagrangian multiplier $\lambda^{\mathbf{t}}$ by $\lambda^{\mathbf{t+1}}=\mathbf{max}\left\{\lambda^{\mathbf{t}}+\theta_{\mathbf{t}}\times s^{\mathbf{t}},0\right\}$

Step 6: If new $\lambda^{\mathbf{t+1}}$ does not have significant change, then we re-estimate $\lambda^{\mathbf{t+1}}$ by modifies step-size $\theta_{\mathbf{t}}$ through $\mu_{\mathbf{t}}=\mu_{\mathbf{t}}\times 2^{-1}$

Step 7: If $\left\|\lambda^{\mathbf{t+1}}-\lambda^{\mathbf{t}}\right\|<\varepsilon$ or subgradient $\mathbf{s}^{\mathbf{t}}=0$, then we have already gained the ideal Lagrange multiplier $\lambda^{\mathbf{t+1}}$, and stop the algorithm. Otherwise, repeat Step 2 to Step 6

Fig. 3 Algorithm 1: The generation of Lagrangian parameter λ_2.

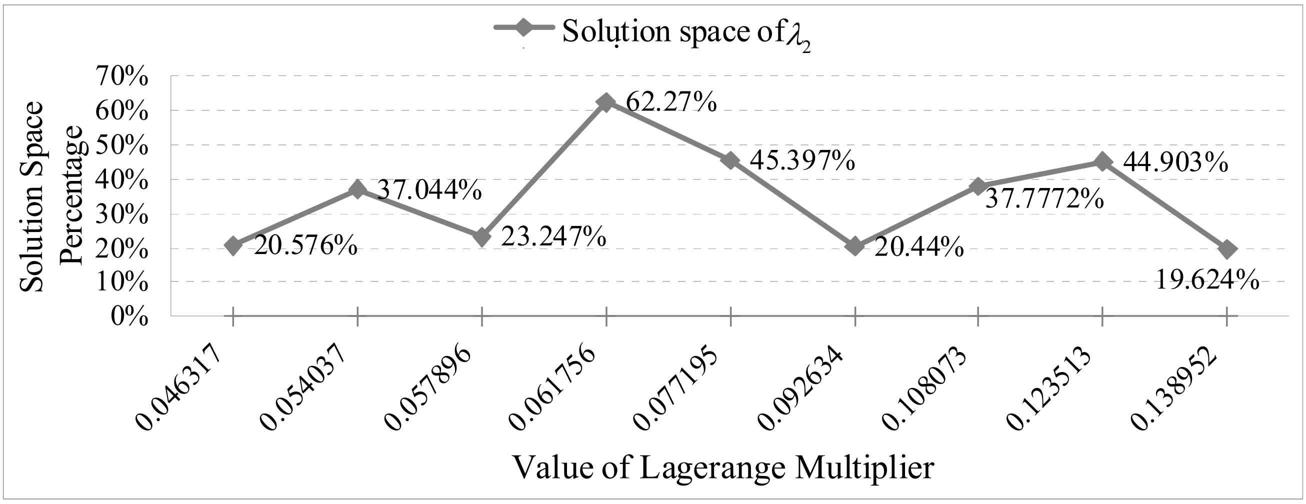

Fig. 4 Quality of Lagrange Multiplier λ_2 .

Table 1 The execution time of ILPF-for-RPS and Lagrange relaxation ILPF-for-RPS with Lagrange Multiplier $\lambda_2 = 0.061756$

Function	$Z_{IP(RPS)}$	$Z_{LR(RPS)}$	$Z_{IP(RPS)}$	$Z_{LR(RPS)}$	$Z_{IP(RPS)}$	$Z_{LR(RPS)}$	$Z_{IP(RPS)}$	$Z_{LR(RPS)}$	$Z_{IP(RPS)}$	$Z_{LR(RPS)}$
Number of Nodes	9		11		13		15		17	
Timeslots	12/16	12/16	14/21	14/21	25/32	25/32	24/25	24/25	xx/44	27/44
Execution Time(m:s)	07:50	00:10	28:52	02:49	52:31	15:48	316:04	11:04	Over 3600:00	31:36
Feasible Solution Probability	100%	100%	100%	100%	100%	100%	100%	100%	xx%	xx%

These simulation results imply that our Lagrange relaxation ILPF-for-RPS can reduce the solution-finding time significantly.

VI. Conclusions

In this paper, we have studied the RPS problem in a mesh network. The RPS problem has been proven to be NP-complete and an ILPF for its optimal solutions has been proposed. However, the existing ILPF-for-RPS has a heavy execution time. This makes the finding of the optimal solutions of the RPS problem impractical in most situations. In this paper, the Lagrange relaxation method has been applied to shrink the solution space of the ILPF-for-RPS for the purpose of shortening the solution-searching time. We have transformed the ILPF-for-RPS into a Lagrange relaxation ILPF-for-RPS. Further, we have proved the objective function of our Lagrange relaxation ILPF-for-RPS to be concave. Computer simulation results show that, in contrast to the original ILPF-for-RPS, our Lagrange relaxation ILPF-for-RPS can decrease the running time by more than 90% in most cases. To sum up, our Lagrange relaxation ILPF-for-RPS simplifies the original ILPF-for-RPS and can be used to find an optimal solution for the RPS problem within a shorter time.

Acknowledgements

This work was supported in part by the National Science Council of the Republic of China under Grant No. NSC 96-2221-E-224-005-MY2.

References

[1] Z. Abichar, Y. Peng, and J. M. Chang, “WiMAX: The emergence of wireless broadbnad”, IT Professional, Vol. 8, Issue 4, pp.44-48, July-August 2006.

[2] A. K. Das, R. J. Marks, M. El-Sharkawi, P. Arabshahi, and A. Gray, “Minimum power broadcast trees for wireless networks: integer programming formulations,” Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM 2003), Vol. 2, pp.1001-1010, 2003.

[3] A. K. Das, R. J. Marks, M. A. El-Sharkawi, P. Arabshahi, and A. Gray, “Optimization methods for minimum power multicasting in wireless networks with sectored antennas,” Proc. of IEEE Wireless Communications and Networking Conference, pp.1299-1304, March 2004.

[4] D. Ghosh, A. Gupta, and P. Mohapatra, “Scheduling in multihop WiMAX Networks,” ACM SIGMOBILE Mobile Computing and Communications Review, Vol. 12, Issue 2, pp.1-11, April 2008.

[5] S. Guo and O. W. Yang, “Minimum-energy multicast routing in static wireless ad hoc networks,” Proc. of IEEE International Conference on Network Protocols (ICNP'04), Vol. 6, pp.3989-3993, September 2004.

[6] M. Held and R. M. Karp, “The traveling salesman problem and minimum spanning trees,” Institute of Operations Research (INFORMS 1970), Vol. 18, pp.1138-1162, 1970.

[7] M. Held and R. M. Karp, “The traveling salesman problem and minimum spanning trees: part II,” Proceedings of the 7th Mathematical Programming Symposium, Vol. 1, No. 1, pp.6-25, December 1971.

[8] F. S. Hillier and G. J. Lieberman, “Introduction to Mathematical Programming,” 2nd ed., McGraw-Hill, 1995.

[9] F. Jin, A. Arora, J. Hwang, and H.-A. Choi, “Routing and packet scheduling for throughput maximization in IEEE 802.16 mesh networks,” Proceedings of IEEE Broadnets, September 2007.

[10] H. Shetiya and V. Sharma, “Algorithms for routing and centralized scheduling to provide QoS in IEEE 802.16 mesh networks,” Proceedings of the 1th ACM Workshop on Wireless Multimedia Networking and Performance Modeling (WMuNeP’05), New York, NY, USA, pp.140-149, October 2005.

[11] H. Shetiya and V. Sharma, “Algorithms for routing and centralized scheduling in IEEE 802.16 mesh networks,” Proceedings of Wireless Communications and Networking Conference (WCNC 2006), pp.147-152, April 2006.

[12] J J. Tao, F. Liu, Z. Zeng, and Z. Lin, “Throughput enhancement WiMAX mesh networks using concurrent transmission,” Proceedings of 2005 International Conference on Wireless Communications, Networking and Mobile Computing, Vol. 2, pp.871-874, September 2005.

[13] K. Thornburg and A. Hummel, “LINGO 8.0 Tutorial: Introduction to LINGO 8.0,” http://www.icaen.uiowa.edu/~ie166/Private/Lingo.pdf

[14] C. Wang, M. T. Thai, Y. Li, F. Wang, and W. Wu, “Minimum coverage breach and maximum network lifetime in wireless sensor networks,” Global Telecommunications Conference (IEEE GLOBECOM 2007), pp.1118-1123, November 2007.

[15] H.-Y. Wei, S. Ganguly, R. Izmailov, and Z. J. Haas, “Interference-aware IEEE 802.16 WiMAX mesh networks,” Proceedings of 61st IEEE Vehicular Technology Conference (VTC 2005 Spring), Vol. 5, pp.3102-3106, 29 May-1 June 2005.