Computer Communications 111 (2017) 153–164

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Optimizing power allocation in wireless networks: Are the implicit constraints really redundant?[☆]

Xiuhua Li[a], Xiaofei Wang[b,*], Victor C.M. Leung[a]

[a] Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC, V6T 1Z4, Canada
[b] Tianjin Key Laboratory of Advanced Networking, School of Computer Science and Technology, Tianjin University, Tianjin 300350, China

ARTICLE INFO

Article history:
Received 15 November 2016
Revised 6 June 2017
Accepted 8 August 2017
Available online 13 August 2017

Keywords:
Power allocation
Water-filling
Subgradient method
Convergence
Step size

ABSTRACT

The widely considered power constraints on optimizing power allocation in wireless networks, e.g., non-negative individual power and limited sum of all the individual power, imply the constraints where each individual power is not greater than the limited sum. However, the related implicit constraints are generally regarded as redundant for algorithm design in most current studies. In this paper, we explore the question "Are the implicit constraints really redundant?" in the optimization of power allocation especially when using iterative methods (e.g., subgradient method) that have slow convergence speeds. Using the water-filling problem as an illustration, we first derive the structural properties of the optimal solutions based on Karush–Kuhn–Tucker conditions. Then we propose a non-iterative closed-form optimal method and use iterative methods (i.e., bisection method and subgradient method) to solve the problem. Our theoretical analysis shows that the implicit constraints are not redundant, and particularly, their consideration can effectively speed up the convergence of the subgradient method and reduce its sensitivity to the chosen step size. Numerical results for the water-filling problem and another existing power allocation problem demonstrate the effectiveness of considering the implicit constraints.

1. Introduction

Future wireless communication networks are required to support a large number of users with various requirements, especially with the growing demands of multimedia services [1–9]. To fulfill the requirements, radio resource management (RRM) plays an essential role as the system level control of co-channel interference and other radio transmission characteristics in wireless communication systems [10]. RRM involves strategies and algorithms for controlling parameters such as transmit power, user association, beamforming, data rate, handover criteria, modulation scheme and error coding scheme, etc., aiming at maximizing the utilization of the limited radio-frequency spectrum resources and radio network infrastructure [11]. Among these RRM techniques, power allocation optimization is a most important aspect of wireless communication system design and has been well studied in the past few decades.

On one hand, to solve various formulated power allocation problems or other optimization problems in wireless systems, the subgradient method is an iterative first-order method that has been widely used in many studies such as [12–26] and references therein. In [12,13], the subgradient method was used to solve the problem of maximizing the throughput under the constraints of interference power and individual transmit power in cognitive radio networks. In [14], subgradient methods were utilized based on dual decomposition to solve the simultaneous routing and resource allocation problem. In [15], a subgradient solution was developed to compute the maximum rate and the optimal routing strategy to solve the maximum multicast rate problem in the general undirected network model. In [16], a distributed subgradient method was used to solve the problem about how to choose opportunistic routes for users to optimize the total utility or profit of multiple simultaneous users in wireless mesh networks. In [17], distributed subgradient methods were applied to optimize global performance in delay tolerant networks with limited information. In [18], a subgradient solution was proposed to solve the problem of jointly optimizing channel pairing, channel-user assignment and power allocation in a single-relay multiple-access system. In [19], an α-approximation dual subgradient algorithm was proposed to optimize the total utility of multiple users in a load-constrained multihop wireless network. Based on the subgradient method,

[☆] A preliminary version of this work was presented at the *8th EAI International Conference on Ad Hoc Networks* (*AdHocNets*), Ottawa, Canada, Sept. 2016.
[*] Corresponding author.
E-mail addresses: lixiuhua@ece.ubc.ca (X. Li), xiaofeiwang@ieee.org (X. Wang), vleung@ece.ubc.ca (V.C.M. Leung).

http://dx.doi.org/10.1016/j.comcom.2017.08.003
0140-3664/© 2017 Elsevier B.V. All rights reserved.

the study in [20] proposed a distributed optimal data gathering cost minimization framework with concurrent data uploading in wireless sensor networks. With the dual subgradient method, the study in [21] focused on convergence analysis of decentralized min-cost subgraph algorithms for multicasting in coded networks. In [22], the subgradient method was used to solve the joint power and bandwidth allocation problem in an improved amplify and forward cooperative communication scheme. In our previous work [23–26], the subgradient method was also used to design resource allocation or scheduling schemes with different optimization objectives under the network constraints in small cell systems. Though subgradient methods can be operated in a distributed manner, they usually have slow convergence rates and are very sensitive to the chosen iteration step sizes [27,28], which need to be improved to reduce the computation costs and even signaling overhead in wireless networks and to reduce the sensitivity to the chosen step sizes since (1) improper step sizes may not make the subgradient methods converge and (2) proper step sizes are not easy to choose especially when the formulated optimization problems are very complex.

On the other hand, mathematically, the formulated optimization problems of power allocation in wireless systems are generally subject to at least two inequality constraints [10,12–24] on p_n, the transmit power allocated at a base station (BS) for the nth user, e.g., (1) nonnegative: $p_n \geq 0, \forall n$, and (2) limited sum: $\sum_{n=1}^{N} p_n \leq P_{\max}$, where N and $P_{\max}$ respectively denote the total number of users served by the BS and the BS's maximum transmit power. These two power constraints imply another set of (implicit) constraints, i.e., $p_n \leq P_{\max}, \forall n$. However, in most currently studied power allocation optimization problems or other similar optimization problems with the above two inequality constraints, the implicit constraints are regarded as redundant and useless in the design of strategies and algorithms for solving the problems. From the perspective of mathematics, the implicit constraints obviously hold, but are they really redundant in optimization algorithms? To the best of our knowledge, this question is unexplored.

The above motivates us to answer the question "Are the implicit constraints really redundant?" in power allocation optimization especially when using subgradient methods in the solution algorithms. Specifically, we study the water-filling problem as a typical illustration of power allocation optimization. Based on Karush–Kuhn–Tucker (KKT) conditions, we derive the structural properties of the optimal solutions to the water-filling problem and evaluate the performance of the proposed methods with/without the consideration of the implicit constraints. As the extension of our previous work [29], our contributions of this paper are summarized below:

- This paper is the first to explore the question "Are the implicit constraints really redundant?" in power allocation optimization especially when using subgradient methods.
- By illustrating the water-filling problem typical in resource allocation, our theoretical analysis shows that considering the implicit constraints can effectively speed up the convergence of the subgradient methods, reduce the sensitivity to the chosen step size and lead to convergence even when an improper step size is used, while the opposite is true if the implicit constraints are not considered. This finding can be extended to other optimization problems and applied to other iterative methods. Besides, we propose a non-iterative closed-form optimal method and iterative bisection methods.
- Numerical results on the water-filling problem and the power allocation problem for multiuser systems in [30] show that considering the implicit constraints in the algorithm design can effectively improve the performance of the used subgradient methods.

The rest of this paper is organized as follows. In Section 2, we formulate the water-filling problem as an illustration of power allocation. In Section 3, we derive the structural properties of the optimal solutions. In Section 4, algorithms for solving the optimization problem are proposed and analyzed. All the possible cases are theoretically analyzed with some well-designed examples in Section 5. Experiment results are shown to evaluate the performance of the proposed algorithms in Section 6. Finally, Section 7 gives the conclusions.

2. The water-filling problem typical in resource allocation

In this section, we provide a general form of resource allocation problems and illustrate a typical resource allocation problem, i.e., water-filling problem, to explore whether the implicit constraints are really redundant for algorithm design.

2.1. General resource allocation problem

Many existing power allocation or other resource allocation optimization problems can be formulated or transformed in a general form as

$$\max_{\mathbf{p},\mathbf{y}} f(\mathbf{p},\mathbf{y}) \tag{1a}$$

$$s.t. \quad p_n \geq 0, \quad \forall n \in \mathcal{N}, \tag{1b}$$

$$\sum_{n=1}^{N} p_n \leq P_{\max}, \tag{1c}$$

$$g_i(\mathbf{p},\mathbf{y}) \leq 0, \quad \forall i \in \mathcal{I}, \tag{1d}$$

$$\mathbf{y} \in \mathcal{S}_Y, \tag{1e}$$

where N is a given number (e.g., number of users), $\mathcal{N} = \{1, 2, \ldots, N\}$, $\mathcal{I}$ and $\mathcal{S}_Y$ are two given sets about resource constraints; $\mathbf{p} = [p_1, p_2, \ldots, p_N]^T$ and $\mathbf{y}$, respectively, are variable vectors of power and other resource allocations; $f(\mathbf{p},\mathbf{y})$ and $g_i(\mathbf{p},\mathbf{y})$ are, respectively, the given objective function (e.g., sum data rate) and constraint functions w.r.t. $\mathbf{p}$ and $\mathbf{y}$; $P_{\max}$ is a positive constant scalar (e.g., maximum transmit power).

From (1b) and (1c), we can get the implicit constraints as

$$p_n \leq P_{\max}, \quad \forall n \in \mathcal{N}. \tag{2}$$

Remark 1. In existing studies, the same or similar implicit constraints in (2) are usually overlooked and are regarded as redundant.

Whether the problems in (1) are convex or nonconvex, they can be solved with a family of iterative methods (e.g., subgradient methods) to get the optimal or suboptimal solutions.

2.2. Water-filling problem

To explore whether the implicit constraints are redundant, we illustrate a most typical resource allocation optimization problem, i.e., the water-filling problem, which is to maximize the sum of the capacity of users under transmit power constraints [31] and can be formulated as

$$\max_{\mathbf{p}} \sum_{n=1}^{N} \log_2(1+\alpha_n p_n) \tag{3a}$$

$$s.t. \quad p_n \geq 0, \quad \forall n \in \mathcal{N}, \tag{3b}$$

$$\sum_{n=1}^{N} p_n \leq P_{\max}, \tag{3c}$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_N]^T$ is a strictly positive constant vector. Clearly, the water-filling problem in (3) is a simple case of the problems in (1) but does not lose the generality.

Then we incorporate the implicit constraints into the problem in (3) and rewrite it in a standard form as

$$\min_{\mathbf{p}} \quad z = -\sum_{n=1}^{N} \log_2(1 + \alpha_n p_n) \tag{4a}$$

$$s.t. -p_n \le 0, \quad \forall n \in \mathcal{N}, \tag{4b}$$

$$p_n - P_{\max} \le 0, \quad \forall n \in \mathcal{N}, \tag{4c}$$

$$\sum_{n=1}^{N} p_n - P_{\max} \le 0, \tag{4d}$$

which is a strictly convex optimization problem. Thus, its locally optimal solution is also globally optimal and the optimal solution is unique. Moreover, we can get the Lagrangian for the problem in (4) as

$$\begin{aligned} L(\mathbf{p}, \boldsymbol{\lambda}, \boldsymbol{s}, \nu) &= -\sum_{n=1}^{N} \log_2(1 + \alpha_n p_n) + \sum_{n=1}^{N} (\nu - \lambda_n + s_n) p_n \\ -\left(\nu + \sum_{n=1}^{N} s_n\right) P_{\max} &= L_0(\mathbf{p}, \nu) - \sum_{n=1}^{N} \lambda_n p_n + \sum_{n=1}^{N} s_n (p_n - P_{\max}) \\ &= L_1(\mathbf{p}, \boldsymbol{\lambda}, \nu) + \sum_{n=1}^{N} s_n (p_n - P_{\max}), \end{aligned} \tag{5}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^N$, $\boldsymbol{s} \in \mathbb{R}^N$ and $\nu \in \mathbb{R}$ are the nonnegative Lagrange multiplier vectors and scalar for constraints (4b), (4c) and (4d), respectively. Besides, we have

$$L_0(\mathbf{p}, \nu) = -\sum_{n=1}^{N} \log_2(1 + \alpha_n p_n) + \nu \left(\sum_{n=1}^{N} p_n - P_{\max}\right), \tag{6}$$

$$L_1(\mathbf{p}, \boldsymbol{\lambda}, \nu) = -\sum_{n=1}^{N} \log_2(1 + \alpha_n p_n) + \sum_{n=1}^{N} (\nu - \lambda_n) p_n + \nu P_{\max}. \tag{7}$$

Remark 2. In current studies, their corresponding Lagrangian is in the form of either L_0 or L_1, but not L.

Thus, we can get the dual objective as $g(\boldsymbol{\lambda}, \boldsymbol{s}, \nu) = \inf_{\mathbf{p}} L(\mathbf{p}, \boldsymbol{\lambda}, \boldsymbol{s}, \nu)$, and then the dual problem as $\max_{\boldsymbol{\lambda}, \boldsymbol{s}, \nu} g(\boldsymbol{\lambda}, \boldsymbol{s}, \nu)$. Since the problem in (4) is convex, the corresponding duality gap reduces to zero at the optimum.

3. Structural properties of the optimal solutions

In this section, we mainly focus on deriving the structural properties of the optimal solutions to the problem in (4).

According to the KKT conditions [27,28], if a feasible solution $\mathbf{p}^* \in \mathcal{S}_P$ is a local (as well as global) minimizer of the convex optimization problem in (4), then there exist multipliers $(\boldsymbol{\lambda}^*, \boldsymbol{s}^*, \nu^*)$, not all zero, $(\boldsymbol{\lambda}^* \succeq 0, \boldsymbol{s}^* \succeq 0, \nu^* \ge 0)$, such that

$$\frac{\partial L}{\partial p_n} = -\frac{\alpha_n}{(1 + \alpha_n p_n^*) \ln 2} - \lambda_n^* + s_n^* + \nu^* = 0, \ \forall n \in \mathcal{N}, \tag{8}$$

$$\lambda_n^* p_n^* = 0, \quad \lambda_n^* \ge 0, \quad p_n^* \ge 0, \quad \forall n \in \mathcal{N}, \tag{9}$$

$$s_n^* (p_n^* - P_{\max}) = 0, \quad s_n^* \ge 0, \quad p_n^* \le P_{\max}, \quad \forall n \in \mathcal{N}, \tag{10}$$

$$\nu^* \left(\sum_{n=1}^{N} p_n^* - P_{\max}\right) = 0, \quad \nu^* \ge 0, \quad \sum_{n=1}^{N} p_n^* \le P_{\max}. \tag{11}$$

Besides, we have $\frac{\partial L_0}{\partial p_n} = -\frac{\alpha_n}{(1+\alpha_n p_n)\ln 2} + \nu$ and $\frac{\partial L_1}{\partial p_n} = -\frac{\alpha_n}{(1+\alpha_n p_n)\ln 2} - \lambda_n + \nu$ for $\forall n \in \mathcal{N}$.

Define $\mathcal{N}_1 \triangleq \{n | s_n^* > 0, n \in \mathcal{N}\}$, $\mathcal{N}_2 \triangleq \{n | s_n^* = 0, n \in \mathcal{N}\}$, and $\boldsymbol{\omega}^* \triangleq \nu^* \cdot \mathbf{1} + \boldsymbol{s}^* \in \mathbb{R}^N$, where $\mathbf{1} = [1, 1, \ldots, 1]^T \in \mathbb{R}^N$. Thus, we have $\mathcal{N}_1 \cup \mathcal{N}_2 = \mathcal{N}$ and $\mathcal{N}_1 \cap \mathcal{N}_2 = \emptyset$. Specifically, if there exists $n \in \mathcal{N}$ such that $p_n^* = P_{\max}$, then denote the index as $k^\#$, i.e., $p_{k^\#}^* = P_{\max}$; otherwise, $k^\#$ does not exist. We give some remarks for the above KKT conditions and derive some structural properties of the optimal solutions via some theorems below.

Remark 3. Specifically, if $\mathcal{N}_1 \ne \emptyset$, then for $\forall n \in \mathcal{N}_1$, we have $p_n^* = P_{\max}$ according to (10), and thus $p_k^* = 0$ and $s_k^* = 0$ for $\forall k \in \mathcal{N}, k \ne n$ according to (10) and (11). Thus, there exists at most one positive element in $\boldsymbol{s}^*$, i.e., $|\mathcal{N}_1| \le 1$. Besides, if $|\mathcal{N}_1| = 1$, the only element in $\mathcal{N}_1$ is equal to $k^\#$ and we have $\mathcal{N}_2 = \mathcal{N} \setminus \{k^\#\}$. Note that even if $\mathcal{N}_1 = \emptyset$, i.e., $\mathcal{N}_2 = \mathcal{N}$, it is possible that $k^\#$ also exists.

Remark 4. For $\forall n \in \mathcal{N}$, $\lambda_n^* s_n^* \equiv 0$. If there exists any $k \in \mathcal{N}$, such that $\lambda_k^* > 0$ and $s_k^* > 0$, then according to (9) and (10), we can get $p_k^* = 0$ and $p_k^* = P_{\max}$ simultaneously, which is clearly contradictory.

Theorem 1. *With $\boldsymbol{\alpha}$ fixed, if $P_{\max}$ is not fixed and can be adjusted, then the optimal objective z^* is strictly decreasing with the increase of $P_{\max}$.*

Theorem 2. *The optimal multiplier scalar ν^* should satisfy that if $\mathcal{N}_1 = \emptyset$,*

$$\nu^* = \max_{n \in \mathcal{N}} \left\{ \frac{\alpha_n}{(1 + \alpha_n p_n^*) \ln 2} \right\}. \tag{12}$$

Otherwise,

$$\nu^* = \frac{\alpha_n}{(1 + \alpha_n P_{\max}) \ln 2} - s_n^*, n \in \mathcal{N}_1, \ \text{and} \ \nu^* \ge \max_{n \in \mathcal{N}_2} \left\{ \frac{\alpha_n}{\ln 2} \right\}. \tag{13}$$

Proof. See Appendix A. □

Remark 5. Based on Theorem 2, we have $\nu^* > 0$, and thus the optimal solution $\mathbf{p}^*$ satisfies $\sum_{n=1}^{N} p_n^* = P_{\max}$ according to (11).

Remark 6. In terms of $\boldsymbol{\omega}^*$, if $\mathcal{N}_1 = \emptyset$, then we have $\boldsymbol{\omega}^* = \nu^* \cdot \mathbf{1} \succ 0$, which indicates that (8) is reduced to the form of existing studies (i.e., $\frac{\partial L_1}{\partial p_n}$) in this case. Otherwise, based on Remarks 3 and 5, we have $\omega_{k^\#}^* = \nu^* + s_{k^\#}^* > \nu^* > 0$ for $k^\# \in \mathcal{N}_1$, and $\omega_n^* = \nu^* > 0$ for $\forall n \in \mathcal{N} \setminus \{k^\#\}$, which indicates that $\boldsymbol{\omega}^*$ is divided into two positive parts.

Theorem 3. *The optimal solution $\mathbf{p}^*$ and the corresponding multiplier scalar ν^* should satisfy*

$$p_n^* = \min \left\{ \left[\frac{1}{\nu^* \ln 2} - \frac{1}{\alpha_n} \right]^+, P_{\max} \right\}, \quad \forall n \in \mathcal{N} \tag{14}$$

where $[x]^+ \triangleq \max\{x, 0\}$.

Proof. See Appendix B. □

Remark 7. From Theorem 3, in the optimal solutions' closed-form expression in (14), the multiplier vectors $(\boldsymbol{\lambda}^*, \boldsymbol{s}^*)$ can be eliminated, while the multiplier scalar ν^* is dominating. However, $\boldsymbol{\lambda}^*$ is to operate $[\mathbf{x}]^+$ such that the optimal solution $\mathbf{p}^* \succeq 0$, while $\boldsymbol{s}^*$ is to operate $\min\{\mathbf{x}, P_{\max}\}$ such that the optimal solution $\mathbf{p}^* \preceq P_{\max}$. In most works, their corresponding solutions are in the form of $[\mathbf{x}]^+$ and have no operation of $\min\{\mathbf{x}, P_{\max}\}$.

Remark 8. From the proof of Theorem 3 in Appendix B, it is not difficult to get another form of the optimal solution $\mathbf{p}^*$, i.e., $p_n^* = [\frac{1}{\omega^* \ln 2} - \frac{1}{\alpha_n}]^+$, $\forall n \in \mathcal{N}$, which is the common form in most works.

Theorem 4. *If $\alpha_{n_1} \geq \alpha_{n_2}$, $n_1 \in \mathcal{N}$, $n_2 \in \mathcal{N}$, then $p_{n_1}^* \geq p_{n_2}^*$ holds in the optimal solution $\mathbf{p}^*$.*

Remark 9. Let $n_1 \in \mathcal{N}$ and $n_2 \in \mathcal{N}$ be two indices such that $\alpha_{n_1} \geq \alpha_{n_2}$. Specifically, for the optimal solution $\mathbf{P}^*$, if $p_{n_1}^* = 0$, then $p_{n_2}^*$ must also be zero based on Theorem 4.

Theorem 5. *There exists the only $k^\# \in \mathcal{N}$ such that $p_{k^\#}^* = P_{\max}$, if and only if both $k^\# = \arg\max_{\boldsymbol{n} \in \mathcal{N}}\{\alpha_n\}$ and $\max_{\boldsymbol{n} \in \mathcal{N}\setminus\{k^\#\}}\{\alpha_n\} \leq \frac{\alpha_{k^\#}}{1+\alpha_{k^\#}P_{\max}}$ hold.*

Proof. See Appendix C. □

Remark 10. Based on Theorem 2 and 5, we discuss the special case where there exists the only $k^\# \in \mathcal{N}$ such that $p_{k^\#}^* = P_{\max}$ as follow.

- If $\max_{n \in \mathcal{N}\setminus\{k^\#\}}\{\alpha_n\} = \frac{\alpha_{k^\#}}{1+\alpha_{k^\#}P_{\max}}$, then we can get $s_{k^\#}^* = 0$, $\lambda_{k^\#}^* = 0$ and $\nu^* = \frac{\alpha_{k^\#}}{(1+\alpha_{k^\#}P_{\max})\ln 2}$. Thus, we have $s_n^* \equiv 0$ for $\forall n \in \mathcal{N}$, and $\lambda_n^* = \frac{\alpha_{k^\#}}{(1+\alpha_{k^\#}P_{\max})\ln 2} - \frac{\alpha_n}{\ln 2}$ for $\forall n \in \mathcal{N} \setminus \{k^\#\}$. Here, $\mathcal{N}_1 = \emptyset$. Thus, the optimal multipliers ($\boldsymbol{\lambda}^*$, $\boldsymbol{s}^*$, ν^*) are unique in this case.
- If $\max_{n \in \mathcal{N}\setminus\{k^\#\}}\{\alpha_n\} < \frac{\alpha_{k^\#}}{1+\alpha_{k^\#}P_{\max}}$, then we can get $s_{k^\#}^* \geq 0$, $\lambda_{k^\#}^* = 0$ and $\nu^* = \frac{\alpha_{k^\#}}{(1+\alpha_{k^\#}P_{\max})\ln 2} - s_{k^\#}^*$. Thus, we have $s_n^* = 0$ and $\lambda_n^* = \frac{\alpha_{k^\#}}{(1+\alpha_{k^\#}P_{\max})\ln 2} - s_{k^\#}^* - \frac{\alpha_n}{\ln 2}$ for $\forall n \in \mathcal{N} \setminus \{k^\#\}$. Here, if $s_{k^\#}^* > 0$, then $\mathcal{N}_1 = \{k^\#\}$; otherwise, $\mathcal{N}_1 = \emptyset$ as well. Since $s_{k^\#}^*$ can take any value in the set $\left\{s_{k^\#}^* \mid 0 \leq s_{k^\#}^* \leq \frac{\alpha_{k^\#}}{(1+\alpha_{k^\#}P_{\max})\ln 2} - \max_{n \in \mathcal{N}\setminus\{k^\#\}}\{\frac{\alpha_n}{\ln 2}\}\right\}$, the optimal multipliers ($\boldsymbol{\lambda}^*$, $\boldsymbol{s}^*$, ν^*) have multiple choices in this case.

As a result, in this special case, we have $\omega_{k^\#}^* = s_{k^\#}^* + \nu^* = \frac{\alpha_{k^\#}}{(1+\alpha_{k^\#}P_{\max})\ln 2}$, which is a constant. However, $\omega_n^* = s_n^* + \nu^* = \nu^*$ for $\forall n \in \mathcal{N} \setminus \{k^\#\}$ are not constants. Moreover, in the considered special case, ν^* can take any value in the set $\{\nu^* \mid \max_{\boldsymbol{n} \in \mathcal{N}\setminus\{k^\#\}}\{\frac{\alpha_n}{\ln 2}\} \leq \nu^* \leq \frac{\alpha_{k^\#}}{(1+\alpha_{k^\#}P_{\max})\ln 2}\}$.

Theorem 6. *Let $\boldsymbol{\pi}$ be the vector obtained by sorting $\boldsymbol{\alpha}$ in a descending order. Then the number of strictly positive elements in the optimal solution $\mathbf{p}^*$ is*

$$\chi = \max\left\{n \in \mathcal{N} \mid \frac{1}{\pi_n} - \frac{1}{n}\left(\sum_{r=1}^{n} \frac{1}{\pi_r} + P_{\max}\right) < 0\right\}, \tag{15}$$

and then the corresponding optimal multiplier ν^* can be expressed as

$$\nu^* = \begin{cases} \text{any value in } \left[\frac{\pi_2}{\ln 2}, \frac{1}{(\frac{1}{\pi_1}+P_{\max})\ln 2}\right], & \chi = 1, \\ \frac{\chi}{\left(\sum_{r=1}^{\chi} \frac{1}{\pi_r}+P_{\max}\right)\ln 2}, & \chi \in \mathcal{N},\ \chi \geq 2. \end{cases} \tag{16}$$

Proof. See Appendix D. □

Remark 11. From (16) in Theorem 6, ν^* can take the value of $\frac{\chi}{(\sum_{r=1}^{\chi} \frac{1}{\pi_r}+P_{\max})\ln 2}$ for all the possible values of χ, which holds in most works where the implicit constraints in (2) are not considered. However, if the implicit constraints in (2) is considered, ν^* may take multiple values as shown in Remark 10 and Theorem 6.

Remark 12. Specifically, if ν^* is chosen to take the value of the closed-form $\frac{\chi}{(\sum_{r=1}^{\chi} \frac{1}{\pi_r}+P_{\max})\ln 2}$ and can be regarded as a fixed value once χ is directly searched, then we can get the closed-form expression of the optimal solution $\mathbf{p}^*$ as $p_n^* = [\frac{1}{\nu^* \ln 2} - \frac{1}{\alpha_n}]^+$, $\forall n \in \mathcal{N}$. Thus, Theorem 6 provides a simple closed-form method to get the optimal solution $\mathbf{p}^*$. Denote the corresponding method as Direct Search Method (DSM), which does not require iterations. From the above analysis, it is not necessary to consider the implicit constraints in the DSM.

From the above analysis, having no consideration of the implicit constraints in (2) can be regarded as a special case of considering them in this paper, which can be extended to other optimization problems. To get the optimal solution $\mathbf{p}^*$ to the problem in (3), whether the implicit constraints are considered in this paper or not in most works, it is very important to get the optimal multiplier ν^* by using either non-iterative methods (i.e., the proposed DSM) or iterative methods (e.g., subgradient method and bisection method). We will show that considering the implicit constraints can greatly improve the convergence rate of iterative methods, especially the subgradient method.

4. Algorithms to solve the optimization problem

In this section, based on the above derived structural properties of the optimal solutions, we provide the DSM, bisection method and subgradient method to solve the problem in (3). Specifically, in terms of the iterative methods, the subgradient methods are used in two scenarios, i.e., considering no implicit constraints and considering implicit constraints.

4.1. Direct search method

Based on Theorem 6, Remarks 11 and 12, the DSM is proposed and described in Algorithm 1. The computation complexity of Algorithm 1 is bounded by the sorting of $\boldsymbol{\alpha}$, i.e., $O(N\log(N))$.

Algorithm 1 Direct Search Method for solving the problem.

```
1: Input: $\boldsymbol{\alpha}$, $P_{\max}$.
2: Initialize $\mathbf{p} = \mathbf{0}_{N\times 1}$, $\chi = 1$, $\nu = 0$.
3: Sort $\boldsymbol{\alpha}$ into $\boldsymbol{\pi}$ in a descending order, and label the index $k$ such
   that $\alpha_k = \pi_1$.
4: Find $\chi$ according to (15).
5: if $\chi = 1$ then
6:     Set $p_k = P_{\max}$.
7:     $\nu$ takes any value in $\left[\frac{\pi_2}{\ln 2}, \frac{1}{(\frac{1}{\pi_1}+P_{\max})\ln 2}\right]$.
8: else
9:     Calculate $\nu = \frac{\chi}{\left(\sum_{r=1}^{\chi} \frac{1}{\pi_r}+P_{\max}\right)\ln 2}$.
10:    Calculate $\mathbf{p}$ as $p_n^* = \left[\frac{1}{\nu^* \ln 2} - \frac{1}{\alpha_n}\right]^+$, $\forall n \in \mathcal{N}$.
11: end if
12: Output: $\mathbf{p}$, $\nu$.
```

4.2. Bisection method

To solve the problem in (3) with bisection method, the optimal multiplier ν^* should be first obtained. Define two monotonic decreasing and continuous functions as

$$f_1(\nu) = \sum_{n=1}^{N}\left[\frac{1}{\nu \ln 2} - \frac{1}{\alpha_n}\right]^+ - P_{\max}, \quad \nu > 0, \tag{17}$$

and

$$f_2(\nu) = \sum_{n=1}^{N} \min\left\{\left[\frac{1}{\nu \ln 2} - \frac{1}{\alpha_n}\right]^+, P_{\max}\right\} - P_{\max}, \quad \nu > 0. \tag{18}$$

Clearly, we have $f_1(\nu) \geq f_2(\nu), \forall \nu > 0$. Denote $\nu_1^* > 0$ and $\nu_2^* > 0$ as the roots of $f_1(\nu) = 0$ and $f_2(\nu) = 0$, respectively. Then both ν_1^* and ν_2^* can be obtained by using the bisection method as shown in Algorithm 2 (denoted by Alg2) and Algorithm 3 (denoted by Alg3), respectively. Alg2 does not consider the implicit constraints, and its form is in common with the algorithms that apply the bisection method to get the optimal multiplier ν^* in most works. Alg3 is the proposed bisection method that considers the implicit constraints.

Algorithm 2 Bisection Method for solving the problem without implicit constraints.

```
1: Input: $\boldsymbol{\alpha}$, $P_{\max}$.
2: Initialize $\mathbf{P} = \mathbf{0}_{N\times 1}$, $\nu = 0$, $\nu_L$, $\nu_M = 0$, $\nu_U$, convergence accuracy $\eta = 10^{-5}$.
3: while not converge do
4:     Set $\nu_M \leftarrow \frac{\nu_L+\nu_U}{2}$.
5:     Check the convergence condition: $|f_1(\nu_M)| < \eta$.
6:     if $f_1(\nu_M) < 0$ then
7:         Set $\nu_L \leftarrow \nu_L$.
8:         Set $\nu_U \leftarrow \nu_M$.
9:     else
10:        Set $\nu_L \leftarrow \nu_M$.
11:        Set $\nu_U \leftarrow \nu_U$.
12:    end if
13: end while
14: Set $\nu \leftarrow \nu_M$.
15: Update $\mathbf{P}$ as $p_n = \left[\frac{1}{\nu \ln 2} - \frac{1}{\alpha_n}\right]^+$, $\forall n \in \mathcal{N}$.
16: Output: $\mathbf{P}$, $\nu$.
```

Algorithm 3 Bisection Method for solving the problem with implicit constraints.

```
1: Input: $\boldsymbol{\alpha}$, $P_{\max}$.
2: Initialize $\mathbf{P} = \mathbf{0}_{N\times 1}$, $\nu = 0$, $\nu_L$, $\nu_M = 0$, $\nu_U$, convergence accuracy $\eta = 10^{-5}$.
3: while not converge do
4:     Set $\nu_M \leftarrow \frac{\nu_L+\nu_U}{2}$.
5:     Check the convergence condition: $|f_2(\nu_M)| < \eta$.
6:     if $f_2(\nu_M) < 0$ then
7:         Set $\nu_L \leftarrow \nu_L$.
8:         Set $\nu_U \leftarrow \nu_M$.
9:     else
10:        Set $\nu_L \leftarrow \nu_M$.
11:        Set $\nu_U \leftarrow \nu_U$.
12:    end if
13: end while
14: Set $\nu \leftarrow \nu_M$.
15: Update $\mathbf{P}$ as $p_n = \min\left\{\left[\frac{1}{\nu \ln 2} - \frac{1}{\alpha_n}\right]^+, P_{\max}\right\}$, $\forall n \in \mathcal{N}$.
16: Output: $\mathbf{P}$, $\nu$.
```

Moreover, in terms of the initialization of the lower bound ν_L and the upper bound ν_U in Alg2 and Alg3, they should satisfy $f_1(\nu_L) \geq f_2(\nu_L) \geq 0$ and $f_2(\nu_U) \leq f_1(\nu_U) \leq 0$. Define $\alpha_{\min} = \min_{\mathbf{n}\in\mathcal{N}}\{\alpha_n\}$ and $\alpha_{\max} = \max_{\mathbf{n}\in\mathcal{N}}\{\alpha_n\}$. Here, we provide a simple method to initialize ν_L and ν_U as follow.

- Set $\nu_L = \frac{N\cdot\alpha_{\min}}{(N+\alpha_{\min}\cdot P_{\max})\ln 2}$. Then we can get $\frac{1}{\nu_L \ln 2} - \frac{1}{\alpha_n} = \frac{P_{\max}}{N} + \frac{1}{\alpha_{\min}} - \frac{1}{\alpha_n} \geq \frac{P_{\max}}{N}$ for $\forall n \in \mathcal{N}$, and thus $f_1(\nu_L) \geq f_2(\nu_L) \geq 0$.
- Set $\nu_U = \frac{N\cdot\alpha_{\max}}{(N+\alpha_{\max}\cdot P_{\max})\ln 2}$. Then we can get $\frac{1}{\nu_U \ln 2} - \frac{1}{\alpha_n} = \frac{P_{\max}}{N} + \frac{1}{\alpha_{\max}} - \frac{1}{\alpha_n} \leq \frac{P_{\max}}{N}$ for $\forall n \in \mathcal{N}$, and thus $f_2(\nu_U) \leq f_1(\nu_U) \leq 0$.

Most importantly, if and only if $k^\# = \arg\max_{\mathbf{n}\in\mathcal{N}}\{\alpha_n\}$ and $\max_{\mathbf{n}\in\mathcal{N}\setminus\{k^\#\}}\{\alpha_n\} < \frac{\alpha_{\max}}{1+\alpha_{\max}P_{\max}}$ hold, i.e., ν^* has multiple values, then Alg3 may converge with less iterations than Alg2, which indicates that considering the implicit constraints can enhance the convergence rate of the bisection method in this case. Otherwise, i.e., ν^* is unique, Alg2 and Alg3 are equivalent and require an identical number of iterations to converge, which indicates that considering the implicit constraints is not necessary.

4.3. Subgradient method

Based on the above analysis, the subgradient method without considering the implicit constraints and the proposed subgradient method with consideration of the implicit constraints are described in Algorithms 4 and 5, respectively. Once the convergence condition is given, the only difference between Algorithms 4 and 5 is the updating of $\mathbf{p}^{(t)}$ at each iteration, i.e., Algorithm 4 has no operation of $\min\{\mathbf{x}, P_{\max}\}$ while the proposed Algorithm 5 has this operation. Note that the structure of Algorithm 4 is in common with the algorithms that apply the subgradient method in most works, and while $\mathbf{p}^*$ is unique, ν^* may not be unique.

Algorithm 4 Subgradient Method for solving the problem without implicit constraints.

```
1: Input: $\boldsymbol{\alpha}$, $P_{\max}$.
2: Initialize $t = 0$, $\mathbf{p}^{(0)} = \mathbf{0}_{N\times 1}$, $\nu^{(0)} = 0.1$, accuracy $\eta = 10^{-5}$.
3: while not converge do
4:     Update $\mathbf{p}^{(t)}$ as $p_n^{(t)} = \left[\frac{1}{\nu^{(t)} \ln 2} - \frac{1}{\alpha_n}\right]^+$, $\forall n \in \mathcal{N}$.
5:     Check convergence condition: $\left|\nu^{(t)}\left(\sum_{n=1}^{N} p_n^{(t)} - P_{\max}\right)\right| < \eta$.
6:     Set $t \leftarrow t + 1$.
7:     Update $\nu^{(t)} = \left[\nu^{(t-1)} + \theta^{(t)}\left(\sum_{n=1}^{N} p_n^{(t-1)} - P_{\max}\right)\right]^+$.
8: end while
9: Output: $\mathbf{p}$, $\nu$.
```

Algorithm 5 Subgradient Method for solving the problem with implicit constraints.

```
1: Input: $\boldsymbol{\alpha}$, $P_{\max}$.
2: Initialize $t = 0$, $\mathbf{p}^{(0)} = \mathbf{0}_{N\times 1}$, $\nu^{(0)} = 0.1$, accuracy $\eta = 10^{-5}$.
3: while not converge do
4:     Update $\mathbf{p}^{(t)}$ as $p_n^{(t)} = \min\left\{\left[\frac{1}{\nu^{(t)} \ln 2} - \frac{1}{\alpha_n}\right]^+, P_{\max}\right\}$, $\forall n \in \mathcal{N}$.
5:     Check convergence condition: $\left|\nu^{(t)}\left(\sum_{n=1}^{N} p_n^{(t)} - P_{\max}\right)\right| < \eta$.
6:     Set $t \leftarrow t + 1$.
7:     Update $\nu^{(t)} = \left[\nu^{(t-1)} + \theta^{(t)}\left(\sum_{n=1}^{N} p_n^{(t-1)} - P_{\max}\right)\right]^+$.
8: end while
9: Output: $\mathbf{p}$, $\nu$.
```

Besides, note that the achieved nonnegative solution **p** may be infeasible in the iteration process with the subgradient method. Then we sketch the proof that considering the implicit constraints can improve the convergence rate of the subgradient method as follow.

1) If there exists $k < +\infty$ such that $\nu^{(k)} = 0$ in the iteration process, we have

- Without consideration of the implicit constraints, $p_n^{(k)} = [\frac{1}{\nu^{(k)}\ln 2} - \frac{1}{\alpha_n}]^+ = +\infty, \forall n \in \mathcal{N}$, and thus $\nu^{(k+1)} = [\nu^{(k)} + \theta^{(k+1)}(\sum_{n=1}^{N} p_n^{(k)} - P_{\max})]^+ = +\infty$. Then we have $p_n^{(k+i)} = 0$, $\forall n \in \mathcal{N}$ and $\nu^{(k+i)} = +\infty$ for all $1 \leq i < +\infty$, which means that the iteration process will not converge in a finite number of iterations.
- With the implicit constraints considered, we have $p_n^{(k)} = P_{\max}$, $\forall n \in \mathcal{N}$, and thus $\nu^{(k+1)} = \theta^{(k+1)}(N-1)P_{\max}$, which can

avoid the above bad case without consideration of the implicit constraints, and thus guarantee convergence in a finite number of iterations.

2) Otherwise, $\nu^{(t)} > 0$ for $\forall t$ holds. We have $\nu^{(t)} = \nu^{(t-1)} + \theta^{(t)}(\sum_{n=1}^N p_n^{(t-1)} - P_{\max}) > 0$ for $\forall t \geq 1$, and then $|\nu^{(t)} - \nu^{(t-1)}| = \theta^{(t)}|\sum_{n=1}^N p_n^{(t-1)} - P_{\max}|$, which indicates that the convergence rate of $\nu^{(t)}$ depends on the steps $\theta^{(t)}$ and the value of $|\sum_{n=1}^N p_n^{(t-1)} - P_{\max}|$. Here, how to choose a proper specific series of steps $\theta^{(t)}$ can be referred to [27,28] and is out of the scope of this paper. Since $|\sum_{n=1}^N \min\{[\frac{1}{\nu^{(t)}\ln 2} - \frac{1}{\alpha_n}]^+, P_{\max}\} - P_{\max}| \leq |\sum_{n=1}^N [\frac{1}{\nu^{(t)}\ln 2} - \frac{1}{\alpha_n}]^+ - P_{\max}|$ for $\forall t$, with the same steps $\theta^{(t)}$, the subgradient method with consideration of the implicit constraints, i.e., Algorithm 5, can achieve a higher convergence rate and needs a smaller number of iterations than the subgradient method without considering the implicit constraints, i.e., Algorithm 4.

In terms of the step size $\theta^{(t)}$, we mainly consider three common categories as follow.

- C1: constant step size. Here, we use $\theta^{(t)} = 0.1$ for $\forall t$.
- C2: nonsummable diminishing step size, i.e., $\theta^{(t)} \geq 0$, $\lim_{t\to\infty}\theta^{(t)} = 0$ and $\sum_{t=1}^{\infty}\theta^{(t)} = \infty$. Here, we use $\theta^{(t)} = \frac{1}{\sqrt{t}}$ for $\forall t$.
- C3: square summable but not summable step size, i.e., $\theta^{(t)} \geq 0$, $\sum_{t=1}^{\infty}[\theta^{(t)}]^2 < \infty$ and $\sum_{t=1}^{\infty}\theta^{(t)} = \infty$. Here, we use $\theta^{(t)} = \frac{100}{t+100N}$ for $\forall t$.

As shown in [27,28], for the constant step size in C1, the subgradient method converges to the optimal value within a small range, i.e., $\lim_{t\to\infty}|\nu^{(t)} - \nu^*| < \varepsilon$. This indicates that the subgradient method finds an $\varepsilon-$suboptimal point within a finite number of iterations. The value ε is a decreasing function of the step size. Moreover, for the diminishing step size in C2 and C3, the subgradient method is guaranteed to converge to the optimal value if the chosen steps are small enough. To simplify the description, denote Alg4-C1, Alg4-C2 and Alg4-C3 as Algorithm 4 with C1, C2 and C3, respectively. Similarly, denote Alg5-C1, Alg5-C2 and Alg5-C3 as Algorithm 5 with C1, C2 and C3, respectively.

Moreover, in terms of different **proper** initializations, both Algorithms 4 and 5 always converge but need different numbers of iterations. Most importantly, in each iteration, the found solution may be infeasible. If the implicit constraints are considered in each iteration, the iterative search of optimal solution can be accelerated by shortening the search subspace of solutions especially for infeasible solutions. Thus, Algorithm 5 can yield a higher convergence rate than Algorithm 4 since the implicit constraints are considered in the former while they are not in the latter.

In terms of the subgradient methods' sensitivity to the chosen step sizes, Algorithm 4 (that does not consider the implicit constraints) may not converge if the step sizes are **improper**. However, for the **improper** step sizes for Algorithm 4, Algorithm 5 (that considers the implicit constraints) can still converge. In Section 6 we will provide some numerical examples showing that considering the implicit constraints leads to convergence even using **improper** step sizes, whereas convergence is not possible if the implicit constraints are not considered.

5. Case study

To explore whether the implicit constraints in (2) are redundant or not to solve the problem especially with subgradient methods, all the possible cases will be theoretically studied with some well-designed examples in this section.

Sort $\boldsymbol{\alpha}$ into $\boldsymbol{\pi}$ in a descending order, i.e., $\pi_1 \geq \pi_2 \geq \ldots \geq \pi_N$, and denote $k \in \mathcal{N}$ such that $\alpha_k = \pi_1$. Considering the implicit constraints, based on the derived structural properties of the optimal solution in Section 3, all the possible cases can be divided into three scenarios as follow.

- S1: $\pi_2 < \frac{\pi_1}{1+\pi_1 P_{\max}}$. In S1, we can get the optimal solution $\mathbf{p}^*$ as $p_k^* = P_{\max}$ and $p_n^* = 0$ for $\forall n \in \mathcal{N}, n \neq k$. Besides, the optimal multiplier ν^* has multiple choices and can take any value in $[\frac{\pi_2}{\ln 2}, \frac{\pi_1}{(1+\pi_1 P_{\max})\ln 2}]$.
- S2: $\pi_2 = \frac{\pi_1}{1+\pi_1 P_{\max}}$. In S2, we can also get the optimal solution $\mathbf{p}^*$ as $p_k^* = P_{\max}$ and $p_n^* = 0$ for $\forall n \in \mathcal{N}, n \neq k$. However, the optimal multiplier $\nu^* = \frac{\pi_1}{1+\pi_1 P_{\max}}$ is unique.
- S3: $\pi_2 > \frac{\pi_1}{1+\pi_1 P_{\max}}$. In S3, we get $p_n^* < P_{\max}$ for $\forall n \in \mathcal{N}$. The optimal multiplier ν^* is unique and can be obtained with either the DSM or the above subgradient methods.

In S1, we provide two numerical examples as follow.

- **Example 1.1**: $N = 3$, $P_{\max} = 1$, $\boldsymbol{\alpha} = [0.4, 2, 0.5]^T$. Thus, we can get the optimal solution $\mathbf{p}^* = [0, 1, 0]^T$, the optimal objective $z^* = -\sum_{n=1}^N \log_2(1+\alpha_n p_n^*) \approx -1.584963$, and the optimal multiplier ν^* takes any value in $[\frac{0.5}{\ln 2}, \frac{2}{3\ln 2}]$, i.e., $[0.721348, 0.961797]$.
- **Example 1.2**: $N = 20$, $P_{\max} = 2$, $\boldsymbol{\alpha} = [2, 0.2, \alpha_3, \ldots, \alpha_{20}]^T$, where α_n is a random value in (0, 0.2] for $\forall n \in \mathcal{N}, n \geq 3$. Thus, we can get the optimal solution $\mathbf{p}^* = [2, 0, 0, \ldots, 0]^T$, the optimal objective $z^* = -\sum_{n=1}^N \log_2(1+\alpha_n p_n^*) \approx -2.321928$, and the optimal multiplier ν^* takes any value in $[\frac{0.2}{\ln 2}, \frac{2}{5\ln 2}]$, i.e., $[0.288539, 0.577078]$.

In S2, we also provide two numerical examples as follow.

- **Example 2.1**: $N = 3$, $P_{\max} = 1$, $\boldsymbol{\alpha} = [0.75, 0.5, 3]^T$. Thus, we can get the optimal solution $\mathbf{p}^* = [0, 0, 1]^T$, the optimal objective $z^* = -\sum_{n=1}^N \log_2(1+\alpha_n p_n^*) = -2$, and the optimal multiplier $\nu^* = \frac{3}{4\ln 2} \approx 1.082021$.
- **Example 2.2**: $N = 20$, $P_{\max} = 2$, $\boldsymbol{\alpha} = [2, 0.4, \alpha_3, \ldots, \alpha_{20}]^T$, where α_n is a random value in (0, 0.4] for $\forall n \in \mathcal{N}, n \geq 3$. Thus, we can get the optimal solution $\mathbf{p}^* = [2, 0, 0, \ldots, 0]^T$, the optimal objective $z^* = -\sum_{n=1}^N \log_2(1+\alpha_n p_n^*) \approx -2.321928$, and the optimal multiplier $\nu^* = \frac{2}{5\ln 2} \approx 0.577078$.

In S3, we also provide two numerical examples as follow.

- **Example 3.1**: $N = 3$, $P_{\max} = 1$, $\boldsymbol{\alpha} = [0.75, 2, 3]^T$. Thus, with the DSM, we can get the optimal solution $\mathbf{p}^* = [0, 0.416667, 0.583333]^T$, the optimal objective $z^* = -\sum_{n=1}^N \log_2(1+\alpha_n p_n^*) \approx -2.333901$, and the optimal multiplier $\nu^* \approx 1.573849$.
- **Example 3.2**: $N = 20$, $P_{\max} = 2$, $\boldsymbol{\alpha} = [2, 1.5, \alpha_3, \ldots, \alpha_{20}]^T$, where α_n is a random value in (0, 2] for $\forall n \in \mathcal{N}, n \geq 3$. Thus, with the DSM, we can directly get the optimal solution, the optimal objective and the optimal multiplier as well.

Moreover, to simplify the description, in terms of the feasibility of the achieved solution $\mathbf{p}$ after the iterative process of the subgradient methods stops, we provide three definitions as follow.

1) General feasibility and *generally feasible*: If the solution $\mathbf{p}$ satisfies $p_n \geq 0$, $\forall n \in \mathcal{N}$ and $\sum_{n=1}^N p_n - P_{\max} \leq \frac{\eta}{\nu}$, where η denotes the convergence accuracy of the used methods, then the solution $\mathbf{p}$ has *general feasibility* and is *generally feasible*.

2) Strong feasibility and *strongly feasible*: If the solution $\mathbf{p}$ satisfies (3b) and (3c), i.e., $p_n \geq 0$, $\forall n \in \mathcal{N}$ and $\sum_{n=1}^N p_n - P_{\max} \leq 0$, then the solution $\mathbf{p}$ has *strong feasibility* and is *strongly feasible*.

3) Weak feasibility and *weakly feasible*: If the solution $\mathbf{p}$ satisfies $p_n \geq 0$, $\forall n \in \mathcal{N}$ and $0 < \sum_{n=1}^N p_n - P_{\max} \leq \frac{\eta}{\nu}$, then the solution $\mathbf{p}$ has *weak feasibility* and is *weakly feasible*.

Clearly, both strong feasibility and weak feasibility are general feasibility. As an extension of the currently used definition of feasibility, the defined strong feasibility is actually the current feasibility, while the defined weak feasibility is actually the current

Table 1
Optimal values in Example 1.1.

Algorithms	z	ν	$\|\mathbf{p}\|_1 - P_{\max}$	Feasibility	# Iteration
DSM	−1.584963	[0.721348, 0.961797]	0	Strong	–
Alg2	−1.584965	0.961795	0.3×10^{-5}	Weak	17
Alg3	−1.584963	0.814698	0	Strong	2
Alg4-C1	−1.584953	0.961802	-1.0×10^{-5}	Strong	99
Alg4-C2	−1.584953	0.961803	-1.0×10^{-5}	Strong	531
Alg4-C3	−1.584955	0.961799	-0.8×10^{-5}	Strong	59
Alg5-C1	−1.584969	0.721345	0.9×10^{-5}	Weak	35
Alg5-C2	−1.584955	0.961800	-0.8×10^{-5}	Strong	20
Alg5-C3	−1.584963	0.764452	0	Strong	2

Table 2
Optimal values in Example 1.2.

Algorithms	z	ν	$\|\mathbf{p}\|_1 - P_{\max}$	Feasibility	# Iteration
DSM	−2.321928	[0.288539, 0.577078]	0	Strong	–
Alg2	−2.321931	0.577077	0.5×10^{-5}	Weak	19
Alg3	−2.321928	0.325140	0	Strong	2
Alg4-C1	−2.321920	0.577080	-1.5×10^{-5}	Strong	82
Alg4-C2	−2.321919	0.577081	-1.7×10^{-5}	Strong	1054
Alg4-C3	−2.321920	0.577080	-1.4×10^{-5}	Strong	112
Alg5-C1	−2.321920	0.577080	-1.4×10^{-5}	Strong	36
Alg5-C2	−2.321921	0.577079	-1.2×10^{-5}	Strong	85
Alg5-C3	−2.321920	0.577080	-1.4×10^{-5}	Strong	62

infeasibility but can also be regarded as feasibility due to the satisfaction of convergence conditions in this paper or other works. Specifically, based on Theorem 1 and Remark 5, if the achieved solution with any of the above methods is weakly feasible, then the achieved objective z is slightly less than the optimal objective z^*; otherwise, the achieved objective z is slightly greater than or equal to the optimal objective z^*.

6. Numerical results

In this section, in addition to the bisection methods, we evaluate by simulations the effectiveness of the subgradient methods considering the implicit constraints in solving the water-filling problem and the power allocation problem in [30], in terms of the convergence speed and sensitivity to the chosen step sizes. Note that the only difference between the iterative methods (i.e., bisection method and subgradient method) with and without considering the implicit constraints is whether to have the operation of $\min\{x, P_{\max}\}$ and the computation complexity the operation can be neglected compared to that of the whole algorithm. Thus, we can use the number of iterations required to satisfy the convergence condition as the convergence speed of the respective algorithms [27,28]. Note that in the following, convergence accuracy refers to the gap between the achieved value (e.g., optimization objective value and multiplier value) and its optimal value, and that we use the ratio of the required numbers of iterations of the subgradient methods without/with considering the implicit constraints for the following comparison of convergence speed.

6.1. Comparisons in S1

Tables 1, 2 and Fig. 1 compare the achieved optimal values, and the values of $|z^{(t)} - z^*|$ and $|\nu^{(t)} - \nu^*|$ versus iteration number in S1 with Example 1.1 and Example 1.2.

In Example 1.1, from Table 1, Fig. 1(a) and (b), we can see the following.

- For the bisection method, Alg3 that considers the implicit constraints can get the optimal solution while Alg2 that does not consider the constraints only achieves a weakly feasible solution. Most importantly, Alg3 only needs 2 iterations to converge and is 7.5 times faster than Alg2.
- For the subgradient method, with each of the three categories of step sizes considered, Algorithm 4 (that does not consider the implicit constraints) always obtains strongly feasible solutions, while only Alg5-C1, i.e., Algorithm 5 (that does consider the implicit constraints) with C1, achieves a weakly feasible solution. However, with each of C1, C2 and C3, Algorithm 5 yields higher convergence accuracy and higher convergence rate than Algorithm 4. Specifically, compared with Alg4-C1, Alg5-C1 can speed up convergence by 182.9%. The convergence rate of Alg5-C2 is 24.5 times faster than Alg4-C2. Besides, Alg5-C3 only needs 2 iterations to converge to the optimal solution and is 28.5 times faster than Alg. 4-C3.

In Example 1.2, from Table 2, Fig. 1(c) and (d), we can see the following.

- For the bisection method, Alg3 can also get the optimal solution while Alg2 only achieves a weakly feasible solution. In particular, Alg3 only needs 2 iterations to converge, which is 8.5 times faster than Alg2.
- For the subgradient method, with each of the three categories of step sizes considered, both Algorithms 4 and 5 always achieve strongly feasible solutions, and Algorithm 5 also has a higher or equal convergence accuracy and higher convergence rate than Algorithm 4. Specifically, compared with **Alg4-C1, Alg5-C1** can speed up the convergence by 127.8%. The convergence rate of **Alg5-C2** is 11.4 times higher than that of **Alg4-C2**. Besides, **Alg5-C3** converges about 1.8 times faster than **Alg4-C3**.

6.2. Comparisons in S2

Tables 3, 4 and Fig. 2 compare the achieved optimal values, and the values of $|z^{(t)} - z^*|$ and $|\nu^{(t)} - \nu^*|$ versus iteration number in S2 with Example 2.1 and Example 2.2.

In Example 2.1, from Table 3, Fig. 2(a) and (b), we can see the following.

- For the bisection method, Alg2 and Alg3 achieve identical results since the optimal multiplier ν^* is unique, which agrees with our theoretical analysis.

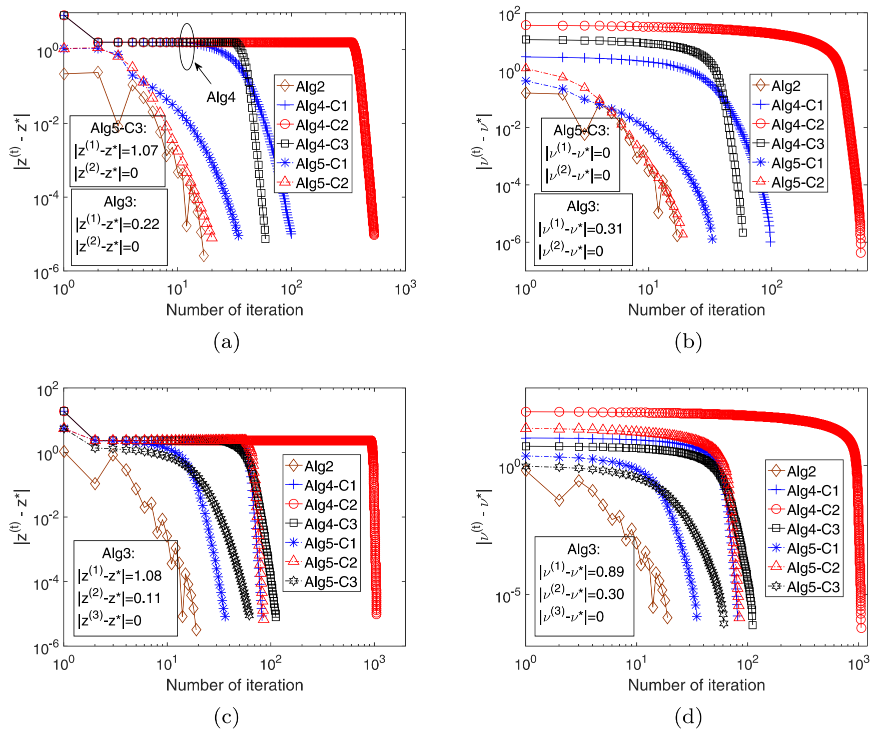


Fig. 1. The values of $|z^{(t)} - z^*|$ and $|\nu^{(t)} - \nu^*|$ versus iteration number in S1, (a) and (b) for Example 1.1, (c) and (d) for Example 1.2.

Table 3
Optimal values in Example 2.1.

Algorithms	z	ν	$\|\mathbf{p}\|_1 - P_{\max}$	Feasibility	# Iteration
DSM	−2	1.082021	0	Strong	–
Alg2	−1.999994	1.082026	-0.6×10^{-5}	Strong	16
Alg3	−1.999994	1.082026	-0.6×10^{-5}	Strong	16
Alg4-C1	−1.999991	1.082027	-0.8×10^{-5}	Strong	123
Alg4-C2	−1.999990	1.082028	-0.9×10^{-5}	Strong	615
Alg4-C3	−1.999993	1.082025	-0.7×10^{-5}	Strong	69
Alg5-C1	−2.000009	1.082015	0.8×10^{-5}	Weak	84
Alg5-C2	−1.999991	1.082026	-0.8×10^{-5}	Strong	31
Alg5-C3	−2.000006	1.082018	0.6×10^{-5}	Weak	23

Table 4
Optimal values in Example 2.2.

Algorithms	z	ν	$\|\mathbf{p}\|_1 - P_{\max}$	Feasibility	# Iteration
DSM	−2.321928	0.577078	0	Strong	–
Alg2	−2.321934	0.577077	1.0×10^{-5}	Weak	19
Alg3	−2.321931	0.577077	0.5×10^{-5}	Weak	19
Alg4-C1	−2.321922	0.577079	-1.1×10^{-5}	Strong	98
Alg4-C2	−2.321918	0.577081	-1.7×10^{-5}	Strong	1616
Alg4-C3	−2.321920	0.577081	-1.5×10^{-5}	Strong	128
Alg5-C1	−2.321922	0.577079	-1.0×10^{-5}	Strong	39
Alg5-C2	−2.321921	0.577080	-1.3×10^{-5}	Strong	110
Alg5-C3	−2.321919	0.577081	-1.5×10^{-5}	Strong	65

- For the subgradient method, with each of the three categories of step sizes considered, Algorithm 4 always obtains strongly feasible solutions, while only **Alg5-C2** achieves a strongly feasible solution. However, with each of C1, C2 and C3, Algorithm 5 has a higher or equal convergence accuracy and higher convergence rate than Algorithm 4. Specifically, **Alg5-C1** is about 1.5 times as fast as **Alg4-C1** while **Alg5-C2** is about 18.8 times faster than **Alg4-C2**. Besides, **Alg5-C3** accelerates the convergence by 200% compared with **Alg4-C3**.

In Example 2.2, from Table 4, Fig. 2(c) and (d), we can see the following.

- For the bisection method, Alg2 and Alg3 also achieve the identical results as the same as our theoretical analysis.
- For subgradient method, with all the considered three categories of step sizes, both Algorithms 4 and 5 always obtain strongly feasible solutions, but Algorithm 5 always outperforms Algorithm 4 in terms of convergence accuracy and convergence rate. In particular, **Alg5-C1, Alg5-C2** and **Alg5-C3** are about 2.5, 14.7 and 2 times as fast as **Alg4-C1, Alg4-C2** and **Alg4-C3**, respectively.

6.3. Comparisons in S3

Tables 5, 6 and Fig. 3 compare the achieved optimal values, and the values of $|z^{(t)} - z^*|$ and $|\nu^{(t)} - \nu^*|$ versus iteration number in S3 with Example 3.1 and Example 3.2.

In Example 3.1, from Table 5, Fig. 3(a) and (b), we can observe the following.

- For bisection method, due to the uniqueness of the optimal multiplier ν^*, Alg2 and Alg3 achieve the same performance in terms of convergence accuracy and convergence rate.
- For subgradient method, Algorithm 4 always obtains strongly feasible solutions for all the considered step sizes, while only **Alg5-C2** achieves a strongly feasible solution. However, with each of C1, C2 and C3, Algorithm 5 has very approximate convergence accuracy and higher convergence rate compared with Algorithm 4. Specifically, **Alg5-C1, Alg5-C2** and **Alg5-C3** are about 1.3, 23.3 and 3.7 times as fast as **Alg4-C1, Alg4-C2** and **Alg4-C3**, respectively.

In Example 3.2, from Table 6, Fig. 3(c) and (d), we can observe the following.

- For bisection method, agreeing with our theoretical analysis, Alg2 and Alg3 also achieve the identical results.
- For subgradient method, both Algorithms 4 and 5 always obtains strongly feasible solutions for all the considered step sizes. However, with each of C1, C2 and C3, Algorithm 5 ahieves

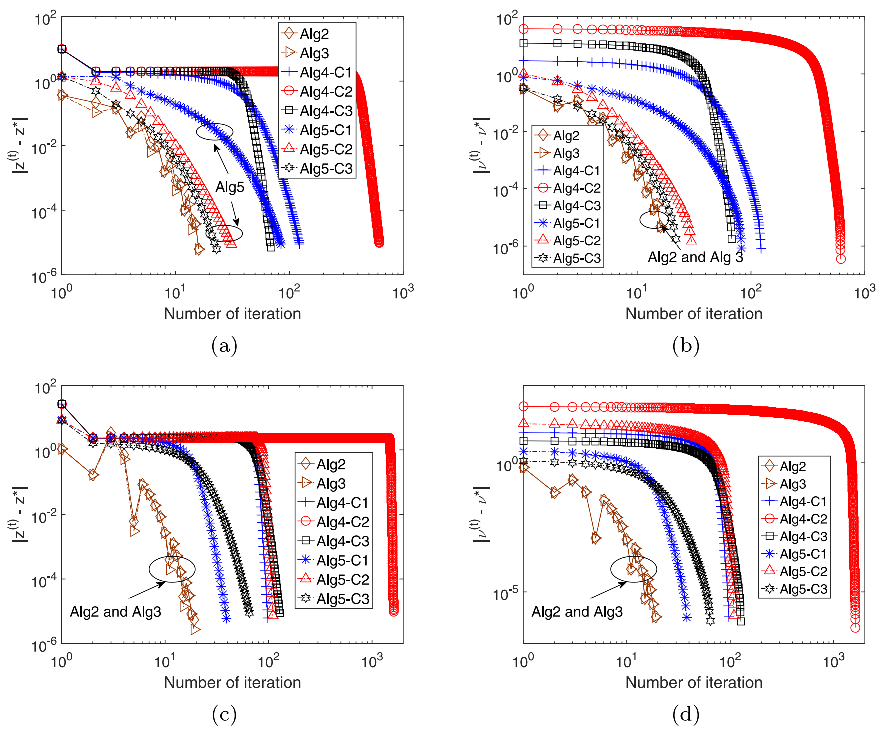


Fig. 2. The values of $|z^{(t)} - z^*|$ and $|\nu^{(t)} - \nu^*|$ versus iteration number in S2, (a) and (b) for Example 2.1, (c) and (d) for Example 2.2.

Table 5
Optimal values in Example 3.1.

Algorithms	z	ν	$\|\mathbf{p}\|_1 - P_{\max}$	Feasibility	# Iteration
DSM	−2.333901	1.573849	0	Strong	–
Alg2	−2.333897	1.573851	-0.2×10^{-5}	Strong	16
Alg3	−2.333897	1.573851	-0.2×10^{-5}	Strong	16
Alg4-C1	−2.333891	1.573854	-0.6×10^{-5}	Strong	123
Alg4-C2	−2.333891	1.573854	-0.6×10^{-5}	Strong	651
Alg4-C3	−2.333892	1.573852	-0.5×10^{-5}	Strong	70
Alg5-C1	−2.333911	1.573844	0.6×10^{-5}	Weak	96
Alg5-C2	−2.333891	1.573853	-0.6×10^{-5}	Strong	28
Alg5-C3	−2.333910	1.573846	0.6×10^{-5}	Weak	26

Table 6
Optimal values in Example 3.2.

Algorithms	z	ν	$\|\mathbf{p}\|_1 - P_{\max}$	Feasibility	# Iteration
DSM	−4.162526	1.781380	0	Strong	–
Alg2	−4.162530	1.781379	0.2×10^{-5}	Weak	19
Alg3	−4.162530	1.781379	0.2×10^{-5}	Weak	19
Alg4-C1	−4.162517	1.781380	-0.5×10^{-5}	Strong	136
Alg4-C2	−4.162516	1.781380	-0.5×10^{-5}	Strong	4211
Alg4-C3	−4.162517	1.781380	-0.5×10^{-5}	Strong	158
Alg5-C1	−4.162516	1.781380	-0.5×10^{-5}	Strong	28
Alg5-C2	−4.162518	1.781380	-0.4×10^{-5}	Strong	114
Alg5-C3	−4.162517	1.781380	-0.5×10^{-5}	Strong	41

higher or equal convergence accuracy and higher convergence rate than Algorithm 4. In particular, **Alg5-C1** is about 4.9 times as fast as **Alg4-C1** while **Alg5-C2** is about 35.9 times faster than **Alg4-C2**. Besides, **Alg5-C3** is about 3.9 times as fast as **Alg4-C3**.

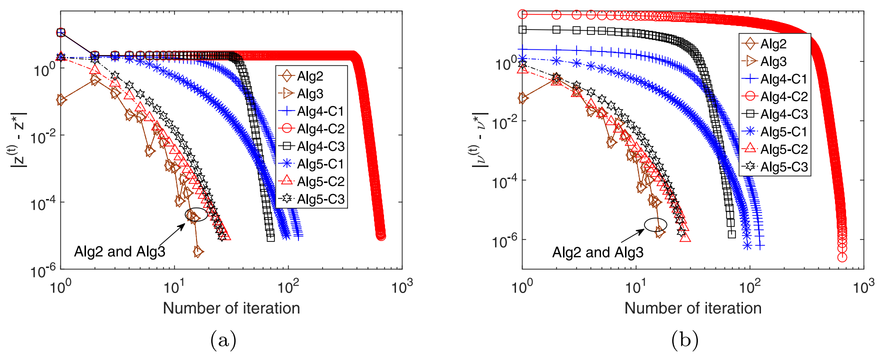
6.4. Comparisons of sensitivity to step size

By setting different step sizes, we show that considering the implicit constraints can reduce the sensitivity to the step size and lead to convergence, while convergence may not be achieved if the implicit constraints are not considered. Here, if the iteration process does not stop within 10^7 iterations, the used method is regarded as not convergent from a practical perspective.

In C1, we set $\theta^{(t)} = 0.9$ for $\forall t$ and see that **Alg4-C1** does not converge in Example 1.2 while **Alg5-C1** converges in 16 iterations. In C2, we set $\theta^{(t)} = \frac{0.01}{\sqrt{t}}$ for $\forall t$ and see that **Alg4-C2** converges in 24, 142 iterations in Example 1.2 while **Alg5-C2** converges in only 2 iterations, which indicates that considering the implicit constraints can speed up convergence by 12,070 times. In C3, we set $\theta^{(t)} = \frac{N}{t+N}$ for $\forall t$ and see that: *(1)* **Alg4-C3** does not converge in Example 1.1 while **Alg5-C3** converges in 34 iterations; *(2)* **Alg4-C3** does not converge in Example 2.1 while **Alg5-C3** converges in 67 iterations; *(3)* **Alg4-C3** does not converge in Example 3.1 while **Alg5-C3** converges in 29 iterations.

6.5. Comparisons for alternate power allocation optimization problem

To further evaluate the effectiveness of considering the implicit constraints, we consider an alternate power allocation optimization problem [30], which maximizes the total system throughput in a multiuser orthogonal frequency-division multiplexing system under the constraints of total power and minimum data rate required by each user. To solve the problem, we use the subgradient method without considering the related implicit constraints (denoted as Alg6) and with this consideration (denoted as Alg7). Note that Alg6 is the subgradient method used in [30]. In our simulations, we set the subcarrier number $N = 10$, user number $K = 4$, total power $P_{tot} = 5$ Watt and all the users' required minimum data rate is 5 bps/Hz. We also use the same channel model as in [30], the same convergence condition as in Algorithms 4 and 5, and the same step sizes in C1, C2 and C3 as in the previous examples. Note that the convergence condition used in this paper is much stricter than that in [30] and thus the algorithms require more iterations to converge but they achieve more accurate solutions. We use the gap between the achieved objective and the optimal objective as the performance metric. All the results are averaged over 100 channel realizations.



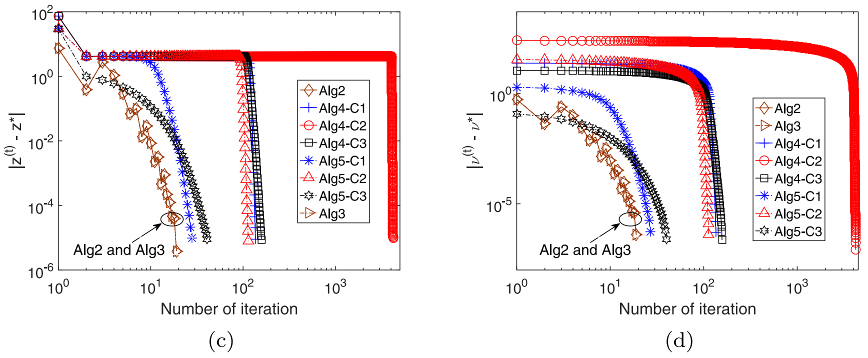


Fig. 3. The values of $|z^{(t)} - z^*|$ and $|\nu^{(t)} - \nu^*|$ versus iteration number in S3, (a) and (b) for Example 3.1, (c) and (d) for Example 3.2.

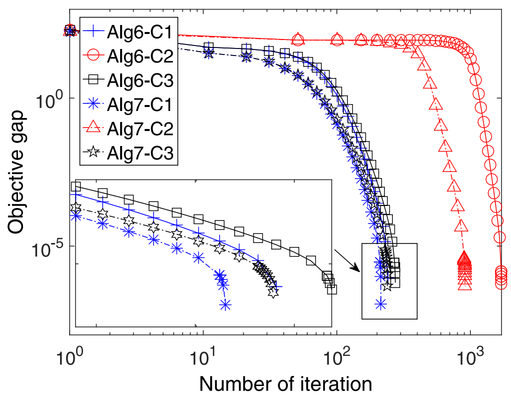


Fig. 4. The values of objective gap versus iteration number.

Fig. 4 compares the performance of the subgradient methods w.r.t. the values of the objective gaps versus iteration number. From Fig. 4, we can observe that Alg7 always has a higher convergence accuracy and higher speed than Alg6. Specifically, in C1, C2 and C3, Alg7 converges about 1.13, 1.87 and 1.15 times as fast as Alg6. Thus, considering the implicit constraints can effectively speed up the convergence of the subgradient method for optimizing the power allocation in [30].

7. Conclusions

In this paper, we have explored the question "Are the implicit constraints really redundant?" in power allocation optimization especially when using subgradient methods. Specifically, by illustrating the water-filling problem to answer the question, we have derived the structural properties of the optimal solutions based on KKT conditions, proposed a non-iterative closed-form optimal method and applied iterative methods (i.e., bisection method and widely used subgradient method) to solve the water-filling problem. Besides, our theoretical analysis has shown that the implicit constraints are not redundant, and their consideration can effectively improve the subgradient methods' convergence speed and reduce the sensitivity to the chosen step sizes in particular. Numerical results have shown that considering the implicit constraints in the water-filling problem can greatly speed up the subgradient methods' convergence speed by up to about 36 times and reduce the sensitivity to the chosen step sizes, and that it can also effectively accelerate its convergence speed by up to 87% in the power allocation problem in [30]. Thus, the implicit constraints are not redundant in the algorithm design. Most importantly, the corresponding theoretical analysis and conclusions about the implicit constraints can be extended to many other resource allocation problems and to other iterative methods.

Acknowledgments

This work is supported in part by a China Scholarship Council Four Year Doctoral Fellowship, the Canadian Natural Sciences and Engineering Research Council through grants RGPIN-2014-06119 and RGPAS-462031-2014 and the National Natural Science Foundation of China through Grant No. 61271182.

Appendix A. Proof of Theorem 2

According to (9), we can get

$$\nu^* = \frac{\alpha_n}{(1+\alpha_n p_n^*)\ln 2} + \lambda_n^* - s_n^*, \quad \forall n \in \mathcal{N}. \tag{A.1}$$

Firstly, we prove the former part in (12) of Theorem 2. If $\mathcal{N}_1 = \emptyset$, we get $s_n^* \equiv 0, \forall n \in \mathcal{N}$. Then based on (A.1), we can get

$$\nu^* = \frac{\alpha_n}{(1+\alpha_n p_n^*)\ln 2} + \lambda_n^*, \quad \forall n \in \mathcal{N}. \tag{A.2}$$

Since $\lambda_n^* \geq 0, \forall n \in \mathcal{N}$, we can obtain $\nu^* \geq \frac{\alpha_n}{(1+\alpha_n p_n^*)\ln 2}$, $\forall n \in \mathcal{N}$. Then we have

$$\nu^* \geq \max_{n \in \mathcal{N}} \left\{ \frac{\alpha_n}{(1+\alpha_n p_n^*)\ln 2} \right\}. \tag{A.3}$$

On the other hand, if $\nu^* > \max_{n\in\mathcal{N}}\{\frac{\alpha_n}{(1+\alpha_n p_n^*)\ln 2}\} > 0$, then according to (A.2), we can get $\lambda_n^* > 0, \forall n \in \mathcal{N}$. Based on (9), we obtain $p_n^* = 0, \forall n \in \mathcal{N}$. Thus, $\sum_{n=1}^{N} p_n^* = 0$. We have $\nu^*(\sum_{n=1}^{N} p_n^* - P_{\max}) = -\nu^* P_{\max} < 0$, which contradicts $\nu^*(\sum_{n=1} p_n^* - P_{\max})$ according to (11). Thus, we can get

$$\nu^* \le \max_{\boldsymbol{n}\in\mathcal{N}} \left\{ \frac{\alpha_n}{(1+\alpha_n p_n^*)\ln 2} \right\}. \tag{A.4}$$

According to (A.3) and (A.4), the proof of the former can be completed.

Then we prove the latter. Since $\mathcal{N}_1 \ne \emptyset$, we have $p_n^* = P_{\max}$ and $s_n^* > 0$, $n \in \mathcal{N}_1$, and $p_k^* = 0, s_k^* = 0, \forall k \in \mathcal{N}_2$ according to Remark 3. Then we get $\lambda_n^* = 0, n \in \mathcal{N}_1$, thus $\nu^* = \frac{\alpha_n}{(1+\alpha_n P_{\max})\ln 2} - s_n^*$, $n \in \mathcal{N}_1$. Besides, we get $\nu^* = \frac{\alpha_n}{\ln 2} + \lambda_n^* \ge \frac{\alpha_n}{\ln 2}$, $\forall n \in \mathcal{N}_2$. Thus, (13) holds.

The whole proof can be completed.

Appendix B. Proof of Theorem 3

According to (8), we can get

$$p_n^* = \frac{1}{(\nu^* - \lambda_n^* + s_n^*)\ln 2} - \frac{1}{\alpha_n}, \quad \forall n \in \mathcal{N}. \tag{B.1}$$

We mainly prove it by discussing two aspects as follow.

1) If there exists $k \in \mathcal{N}$, such that $p_k^* = P_{\max}$, then we can get $p_n^* = 0$ for $\forall n \in \mathcal{N}, n \ne k$. Based on (9) and (10), we can respectively get

$$\begin{cases} \lambda_n^* = 0 \text{ and } s_n^* \ge 0, & n = k, \\ \lambda_n^* \ge 0 \text{ and } s_n^* = 0, & \forall n \in \mathcal{N}, n \ne k. \end{cases} \tag{B.2}$$

In this aspect, we prove it by discussing two cases as follow.

- For k, we have $P_{\max} = p_k^* = \frac{1}{(\nu^*+s_k^*)\ln 2} - \frac{1}{\alpha_k} \le \frac{1}{\nu^*\ln 2} - \frac{1}{\alpha_k}$, and thus $p_k^* = \min\{\frac{1}{\nu^*\ln 2} - \frac{1}{\alpha_k}, P_{\max}\} = \min\{[\frac{1}{\nu^*\ln 2} - \frac{1}{\alpha_k}]^+, P_{\max}\}$.
- For $\forall n \in \mathcal{N}, n \ne k$, we have $0 = p_n^* = \frac{1}{(\nu^*-\lambda_n^*)\ln 2} - \frac{1}{\alpha_n} \ge \frac{1}{\nu^*\ln 2} - \frac{1}{\alpha_n}$, and thus $p_n^* = [\frac{1}{\nu^*\ln 2} - \frac{1}{\alpha_n}]^+ = \min\{[\frac{1}{\nu^*\ln 2} - \frac{1}{\alpha_k}]^+, P_{\max}\}$.

2) Otherwise, that is, $p_n^* < P_{\max}$ for $\forall n \in \mathcal{N}$, we can get $s_n^* = 0, \forall n \in \mathcal{N}$ according to (10). Thus, from (B.1), we can get $p_n^* = \frac{1}{(\nu^*-\lambda_n^*)\ln 2} - \frac{1}{\alpha_n}$, $\forall n \in \mathcal{N}$. Based on (9), for $\forall n \in \mathcal{N}$, we can get

$$\lambda_n^* \begin{cases} \ge 0, & p_n^* = 0, \\ = 0, & p_n^* > 0, \end{cases} \quad \text{and} \quad p_n^* \begin{cases} \ge 0, & \lambda_n^* = 0, \\ = 0, & \lambda_n^* > 0. \end{cases} \tag{B.3}$$

In this aspect, we also prove it by discussing two cases as follow.

- For $\forall n \in \mathcal{N}$ such that $p_n^* = 0$, we have $0 = p_n^* = \frac{1}{(\nu^*-\lambda_n^*)\ln 2} - \frac{1}{\alpha_n} \ge \frac{1}{\nu^*\ln 2} - \frac{1}{\alpha_n}$, and thus $p_n^* = [\frac{1}{\nu^*\ln 2} - \frac{1}{\alpha_n}]^+ = \min\{[\frac{1}{\nu^*\ln 2} - \frac{1}{\alpha_n}]^+, P_{\max}\}$.
- For $\forall n \in \mathcal{N}$ such that $p_n^* > 0$ (also, $p_n^* < P_{\max}$), we have $0 < p_n^* = \frac{1}{\nu^*\ln 2} - \frac{1}{\alpha_n} < P_{\max}$, and thus $p_n^* = \min\{[\frac{1}{\nu^*\ln 2} - \frac{1}{\alpha_n}]^+, P_{\max}\}$.

The whole proof can be completed.

Appendix C. Proof of Theorem 5

The proof consists of two parts, i.e., the necessity and sufficiency. Define $j = \arg\max_{\boldsymbol{n}\in\mathcal{N}\setminus\{k^\#\}}\{\alpha_n\}$.

First, we prove the necessity, i.e., if $k^\# = \arg\ \max_{\boldsymbol{n}\in\mathcal{N}}\{\alpha_n\}$ and $\max_{\boldsymbol{n}\in\mathcal{N}\setminus\{k^\#\}}\{\alpha_n\} \le \frac{\alpha_{k^\#}}{1+\alpha_{k^\#}P_{\max}}$ hold, then $p_{k^\#}^* = P_{\max}$ holds and $k^\#$ is unique. In this case, since $\frac{\alpha_{k^\#}}{1+\alpha_{k^\#}P_{\max}} < \alpha_{k^\#}$, we have $\max_{\boldsymbol{n}\in\mathcal{N}\setminus\{k^\#\}}\{\alpha_n\} = \alpha_j < \alpha_{k^\#}$. As a result, $k^\#$ is unique. Based on Theorem 4, we have $p_{k^\#}^* = \max_{\boldsymbol{n}\in\mathcal{N}}\{p_n^*\}$. Assuming $p_{k^\#}^* < P_{\max}$, then we have $0 < p_j^* \le P_{\max} - p_{k^\#}^*$ based on Remark 5. Thus, based on (9) and (10), we have $\lambda_{k^\#}^* = \lambda_j^* = 0$ and $s_{k^\#}^* = s_j^* = 0$. Then according to (A.1), we have $\nu^* = \frac{\alpha_j}{(1+\alpha_j p_j^*)\ln 2} = \frac{\alpha_{k^\#}}{(1+\alpha_{k^\#} p_{k^\#}^*)\ln 2} > \frac{\alpha_{k^\#}}{(1+\alpha_{k^\#}P_{\max})\ln 2} \ge \frac{\alpha_j}{\ln 2}$. Thus, we have $\frac{\alpha_j}{1+\alpha_j p_j^*} > \alpha_j$, which does not hold. Thus, the assumption does not hold, i.e., $p_{k^\#}^* = P_{\max}$, which proves the necessity.

Then, we prove the sufficiency, i.e., if there exists the only $k^\# \in \mathcal{N}$ such that $p_{k^\#}^* = P_{\max}$, then both $k^\# = \arg\ \max \boldsymbol{n} \in \mathcal{N}\{\alpha_n\}$ and $\max_{\boldsymbol{n}\in\mathcal{N}\setminus\{k^\#\}}\{\alpha_n\} \le \frac{\alpha_{k^\#}}{1+\alpha_{k^\#}P_{\max}}$ hold. In this case, based on Theorem 4, it is clear that $k^\# = \arg\ \max_{\boldsymbol{n}\in\mathcal{N}}\{\alpha_n\}$ holds. Based on Theorem 2, either $\mathcal{N}_1 = \emptyset$ or $\mathcal{N}_1 \ne \emptyset$, $\max_{\boldsymbol{n}\in\mathcal{N}\setminus\{k^\#\}}\{\alpha_n\} \le \frac{\alpha_{k^\#}}{1+\alpha_{k^\#}P_{\max}}$ always holds. Thus, the sufficiency is proved.

The whole proof can be completed.

Appendix D. Proof of Theorem 6

Let $\mathbf{Q}^*$ denote the vector obtained by sorting the optimal solution $\mathbf{P}^*$ in a descending order. According to Theorems 3 and 4, we can get

$$q_n^* = \min\left\{\left[\frac{1}{\nu^*\ln 2} - \frac{1}{\pi_n}\right]^+, P_{\max}\right\}, \quad \forall n \in \mathcal{N}. \tag{D.1}$$

We prove Theorem 6 by discussing two aspects as follow.

- If $\chi = 1$, i.e., $q_1^* = P_{\max}$, then based on Theorem 5 and Remark 10, the corresponding conclusions hold.
- Otherwise, i.e., $q_n^* < P_{\max}$ for $\forall n \in \mathcal{N}$, then based on Remarks 6 and 8, (D.1) is reduced to

  $$q_n^* = \left[\frac{1}{\nu^*\ln 2} - \frac{1}{\pi_n}\right]^+, \quad \forall n \in \mathcal{N}, \chi \ge 2. \tag{D.2}$$

  Thus, we have

  $$q_n^* \begin{cases} = \frac{1}{\nu^*\ln 2} - \frac{1}{\pi_n} > 0, & 1 \le n \le \chi, \\ = 0 \ge \frac{1}{\nu^*\ln 2} - \frac{1}{\pi_n}, & \forall n \in \mathcal{N} \setminus \{1, 2, \ldots, \chi\}. \end{cases} \tag{D.3}$$

  Based on (D.3) and Remark 5, we have $P_{\max} = \sum_{r=1}^{\chi} q_r^* = \frac{\chi}{\nu^*\ln 2} - \sum_{r=1}^{\chi}\frac{1}{\pi_r}$. Thus, we have $\nu^* = \frac{\chi}{(\sum_{r=1}^{\chi}\frac{1}{\pi_r}+P_{\max})\ln 2}$, which indicates (16) holds.

  Moreover, we can get $q_n^* = \frac{\sum_{r=1}^{\chi}\frac{1}{\pi_r}+P_{\max}}{\chi} - \frac{1}{\pi_n} > 0$ for $\forall n \in \mathcal{N}, 1 \le n \le \chi$, and $q_n^* = 0 \ge \frac{\sum_{r=1}^{\chi}\frac{1}{\pi_r}+P_{\max}}{\chi} - \frac{1}{\pi_n}$ for $\forall n \in \mathcal{N} \setminus \{1, 2, \ldots, \chi\}$. Clearly, we have $\frac{1}{\pi_\chi} - \frac{1}{\chi}(\sum_{r=1}^{\chi}\frac{1}{\pi_r} + P_{\max}) < 0$ and $\frac{1}{\pi_{\chi+1}} - \frac{1}{\chi+1}(\sum_{r=1}^{\chi+1}\frac{1}{\pi_r} + P_{\max}) = \frac{\chi}{\chi+1}[\frac{1}{\pi_{\chi+1}} - \frac{1}{\chi}(\sum_{r=1}^{\chi}\frac{1}{\pi_r} + P_{\max})] \ge 0$ if $\chi + 1 \in \mathcal{N}$. Besides, for $\forall j \in \{1, 2, \ldots, N-\chi\}$, we have $\frac{1}{\pi_{\chi+j}} - \frac{1}{\chi+j}(\sum_{r=1}^{\chi+j}\frac{1}{\pi_r} + P_{\max}) = \frac{\chi+j-1}{\chi+j}[\frac{1}{\pi_{\chi+j}} - \frac{1}{\chi+j-1}(\sum_{r=1}^{\chi+j-1}\frac{1}{\pi_r} + P_{\max})] \ge \frac{\chi+j-1}{\chi+j}[\frac{1}{\pi_{\chi+j-1}} - \frac{1}{\chi+j-1}(\sum_{r=1}^{\chi+j-1}\frac{1}{\pi_r} + P_{\max})]$. By applying mathematical induction, we have $\frac{1}{\pi_{\chi+j}} - \frac{1}{\chi+j}(\sum_{r=1}^{\chi+j}\frac{1}{\pi_r} + P_{\max}) \ge 0$ for $\forall j \in \{1, 2, \ldots, N-\chi\}$. Thus, (15) holds.

Based on the above two aspects' analysis, the whole proof can be completed.

References

[1] X. Wang, Z. Sheng, S. Yang, V.C.M. Leung, Tag-assisted social-aware opportunistic d2d sharing for traffic offloading in mobile social networks, IEEE Wirel. Commun. Mag. 23 (4) (2016) 60–67.

[2] X. Wang, X. Li, V.C.M. Leung, P. Nasiopoulos, A framework of cooperative cell caching for the future mobile networks, in: Proceedings of HICSS, 2015, pp. 5404–5413.

[3] X. Li, X. Wang, S. Xiao, V.C.M. Leung, Delay performance analysis of cooperative cell caching in future mobile networks, in: Proceedings of IEEE ICC, 2015, pp. 5652–5657.

[4] X. Li, X. Wang, C. Zhu, W. Cai, V.C.M. Leung, Caching-as-a-service: virtual caching framework in the cloud-based mobile networks, in: Proceedings of IEEE INFOCOM Workshops, 2015, pp. 372–377.

[5] X. Li, X. Wang, K. Li, V.C.M. Leung, Caas: caching as a service for 5g networks, IEEE Access 5 (2017) 5982–5993.

[6] X. Li, X. Wang, V.C.M. Leung, Weighted network traffic offloading in cache-enabled heterogeneous networks, in: Proceedings of IEEE ICC, 2016, pp. 1–6.

[7] X. Li, X. Wang, K. Li, V.C.M. Leung, Collaborative hierarchical caching for traffic offloading in heterogeneous networks, Proceedings of IEEE ICC, 2017.

[8] X. Li, P. Wu, X. Wang, K. Li, Z. Han, V.C.M. Leung, Collaborative hierarchical caching in cloud radio access networks, Proceedings of IEEE INFOCOM Workshops, 2017.

[9] H. Wang, X. Wang, K. Li, J. Ren, X. Zhang, D. Jian, A measurement study of device-to-device sharing in mobile social networks based on spark, 29, Wiley Concurrency and Computation: Practice and Experience, 2017, pp. 1–11. 16.

[10] J. Zander, S.L. Kim, M. Almgren, Radio Resource Management, Artech House Publishers, 2001.

[11] M. Peng, C. Wang, J. Li, H. Xiang, V. Lau, Recent advances in underlay heterogeneous networks: interference control, resource allocation, and self-organization, IEEE Commun. Surv. Tutor. 17 (2) (2015) 700–729. Second Quarter.

[12] L. Zhang, Y. Xin, Y.C. Liang, H.V. Poor, Cognitive multiple access channels: optimal power allocation for weighted sum rate maximization, IEEE Trans. Commun. 57 (9) (2009) 2754–2762.

[13] D.N. Nguyen, M. Krunz, Spectrum management and power allocation in MIMO cognitive networks, in: Proceedings of IEEE INFOCOM, 2012, pp. 2023–2031.

[14] L. Xiao, M. Johansson, S. Boyd, Simultaneous routing and resource allocation via dual decomposition, IEEE Trans. Commun. 52 (7) (2004) 136–1134.

[15] Z. Li, B. Li, Efficient and distributed computation of maximum multicast rates, in: Proceedings of IEEE INFOCOM, 2005, pp. 1618–1628.

[16] X. Fang, D. Yang, G. Xue, Consort: node-constrained opportunistic routing in wireless mesh networks, in: Proceedings of IEEE INFOCOM, 2011, pp. 1907–1915.

[17] R. Masiero, G. Neglia, Distributed subgradient methods for delay tolerant networks, in: Proceedings of IEEE INFOCOM, 2011, pp. 261–265.

[18] M. Hajiaghayi, M. Dong, B. Liang, Optimal channel assignment and power allocation for dual-hop multi-channel multi-user relaying, in: Proceedings of IEEE INFOCOM, 2011, pp. 76–80.

[19] X. Fang, D. Yang, G. Xue, Resource allocation in load-constrained multihop wireless networks, in: Proceedings of IEEE INFOCOM, 2012, pp. 280–288.

[20] S. Guo, Y. Yang, A distributed optimal framework for mobile data gathering with concurrent data uploading in wireless sensor networks, in: Proceedings of IEEE INFOCOM, 2012, pp. 1305–1313.

[21] F. Zhao, M. Médard, A. Ozdaglar, D. Lun, Convergence study of decentralized min-cost subgraph algorithms for multicast in coded networks, IEEE Trans. Inf. Theory 60 (1) (2014) 410–421.

[22] H.A. Tous, I. Barhumi, Resource allocation for multiuser improved AF cooperative communication scheme, IEEE Trans. Wirel. Commun. 14 (7) (2015) 3655–3672.

[23] X. Li, X. Ge, F. Li, V.C.M. Leung, Max-min fair resource allocation for min-rate guaranteed services in distributed antenna systems, in: Proceedings of IEEE VTC-Fall, 2014, pp. 1–5.

[24] X. Li, X. Ge, X. Wang, J. Cheng, V.C.M. Leung, Energy efficiency optimization: joint antenna-subcarrier-power allocation in OFDM-DASs, IEEE Trans. Wirel. Commun. 15 (11) (2016) 7470–7483.

[25] X. Ge, X. Li, H. Jin, J. Cheng, V.C.M. Leung, Joint user association and scheduling for load balancing in heterogeneous networks, in: Proceedings of IEEE GLOBECOM, 2016, pp. 1–6.

[26] X. Li, X. Wang, K. Li, H. Chi, V.C.M. Leung, Resource allocation for content delivery in cache-enabled OFDM small cell networks, Proceedings of IEEE VTC-Fall, 2017.

[27] S. Boyd, L. Vandenbergh, Convex Optimization, Cambridge University Press, 2004.

[28] O. Güler, Foundations of Optimization, Springer Press, 2010.

[29] X. Li, V.C.M. Leung, Optimizing power allocation in wireless networks: Are the implicit constraints really redundant? Proceedings of AdhocNets, Ottawa, Canada, 2016.

[30] Y.J. Chung, C.H. Paik, H.G. Kim, Subgradient approach for resource management in multiuser OFDM systems, in: Proceedings of International Conference on Communications and Electronics, 2006, pp. 203–207.

[31] P. He, L. Zhao, S. Zhou, Z. Niu, Water-filling: a geometric approach and its application to solve generalized radio resource allocation problems, IEEE Trans. Wirel. Commun. 12 (7) (2013). 3637–3467.