2017 American Control Conference
Sheraton Seattle Hotel
May 24–26, 2017, Seattle, USA

Proportional Control Applied to Dynamic Network Resource Allocation

Zhiyuan Liu and Lijun Chen

Abstract— **Many proposed network resource allocation algorithms in existing literature, especially those in practical networking, are designed based on heuristics and then validated by (limited) prototyping and experimentation or numerical simulations. These algorithms usually perform poorly in either responsiveness or convergence, and lack a principled way to trade off between the convergence and the optimality of resource allocation. This paper shows that for a very general resource allocation problem, algorithms based on the proportional control achieve both fast response and convergence. We establish the convergence of the proportional control algorithms with inter-task and spatial couplings, and quantify their performance under estimation error. We further establish the convergence of the algorithms when operating asynchronously with bounded delays. Numerical experiments are provided to complement the theoretical analysis.**

I. Introduction

Resource allocation is an important problem in communication networks. Traditionally, there are two groups of work on network resource allocation. One group of work, mainly by theoretical networking community, tends to formulate well-defined optimization or game theoretic problems for resource allocation and derives algorithms with guaranteed performance. However, these works often ignore practical constraints in real systems, or the resulting algorithms have high implementation complexity that render them impractical. The other group of work, mainly by practical networking community, takes those practical constraints into consideration in resource allocation design. However, the designs are mostly based on heuristics and then validated by limited prototyping and experimentation or numerical simulations. They usually perform poorly in either responsiveness or convergence, and lack a principled way to trade off between the convergence and the optimality of resource allocation.

We have inspected many heuristic resource allocation algorithms such as those in [1], [2], [3]. For instance, reference [1] studies software-defined measurement (SDM) where TCAM entries [4], [5] need to be dynamically allocated to different measurement tasks, and proposes combinations of additive increase (AI), additive decrease (AD), multiplicative increase (MI) and multiplicative decrease (MD) for TCAM allocation to meet the target measurement accuracy. As shown in Fig. 3, these algorithms either have slow response or exhibit oscillatory behavior.

Resource allocation in SDM typifies a general class of network resource allocation problems where resources at one node are shared by different tasks or functions (which we call *inter-task coupling*), a single task or function needs to pool resources from different nodes (which we call *spatial coupling*), and the target allocation is unknown. Inspired by the problem in [1], we describe a general model for network resource allocation, and show that algorithms based on proportional control [6], one of the most basic and simplest control mechanisms, can achieve fast response and convergence in resource allocation.

Specifically, we show that the proportional control algorithm can be seen as a gradient algorithm for solving a quadratic optimization problem. The optimization problem formulation provides a general framework to establish the convergence of the algorithm, quantify its performance under estimation error, as well as establish the convergence of the algorithm when operating asynchronously with bounded delay. We further provide numerical experiments to complement the theoretical analysis.

The rest of the paper is organized as follows. Section II describes the system model. Section III studies the case of a single task. Section IV investigates inter-task coupling in resource allocation. Section V investigates spatial coupling in resource allocation. Section VI studies the convergence of asynchronous algorithm. Section VII presents numerical examples, and Section VIII concludes the paper.

II. System Model

Consider a network as shown in Fig. 1 that consists of a set $\mathcal{N}$ of nodes such as switches in a communication network, each indexed by $j \in \mathcal{N}$. There are a set $\mathcal{M}$ of tasks such as the measurement tasks in SDM running on the network. Each task $i \in \mathcal{M}$ is associated with a certain performance target u_i; e.g., the desired measurement accuracy in SDM. Each task i will require resources from a subset $\mathcal{S}_i$ of nodes, which leads to spatial coupling between different nodes; and each node j contributes resource to a subset $\mathcal{T}_j$ of tasks, which leads to inter-task coupling at the node. Denote by r_{ij} the amount of resource at node j that is allocated to task i. For a given allocation r_{ij}, the task i achieves a certain local performance $l_{ij}(r_{ij})$ at node j; e.g., the local measurement accuracy in the case of SDM. The exact form of $l_{ij}(r_{ij})$ is usually unknown, but the value of l_{ij} for a given r_{ij} can be estimated. We make the following assumptions.

Assumption 1: $l_{ij}(\cdot)$ is a strictly increasing and differentiable function with bounded first- and second-order derivatives, i.e., there exist positive constants γ, μ and κ such that $\gamma \le l'_{ij} \le \mu$ and $|l''_{ij}| \le \kappa, \forall i \in \mathcal{M}, \forall j \in \mathcal{N}$.

Assumption 2: There exists a positive constant ρ such that $u_i, l_{ij} \in [0, \rho], \forall i \in \mathcal{M}, \forall j \in \mathcal{N}$.

Z. Liu and L. Chen are with Department of Computer Science, University of Colorado, Boulder, CO 80309, USA (emails: {zhiyuan.liu, lijun.chen}@colorado.edu).

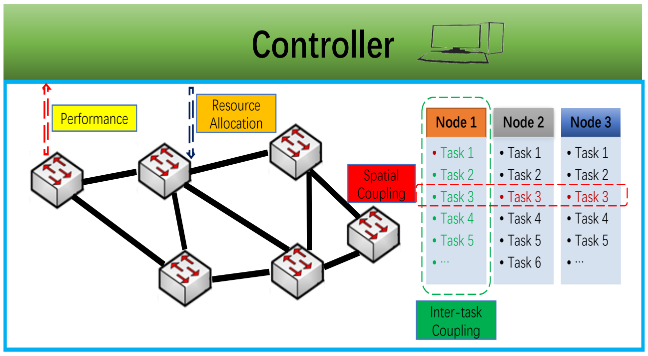


Fig. 1: System Model

Given local performance $\{l_{ij}\}_{j\in\mathcal{S}_i}$, the task i achieves certain global performance $G_i(\{l_{ij}\}_{j\in\mathcal{S}_i})$, where $G_i(\cdot)$ is assumed to be non-decreasing in its arguments. Our goal is to decide on r_{ij} so as to achieve the target performance u_i, i.e., $G_i(\{l_{ij}\}_{j\in\mathcal{S}_i}) = u_i$.

III. Proportional Control Based Resource Allocation for Single Task

In this section we consider the basic problem of achieving the target performance for a single (sub)task so that $l_{ij}(r_{ij}) = u_i$ by incrementally adjusting resource allocation. This can be formulated as a feedback control problem as shown in Fig. 2. At time t, at node j the amount $r_{ij}(t)$ of resource is allocated to task i to achieve certain performance $l_{ij}(t)$. Resource controller then computes $r_{ij}(t{+}1)$ for next time slot $t{+}1$ based on $l_{ij}(t)$ and u_i. Dream [1] takes combinations of AI, AD, MI and MD for dynamic resource allocation based on whether the current performance is above the target or not. As shown in Fig. 3, [1] introduces lots of overshoots and oscillations because it does not take advantage of the information on the gap or distance between the current and target performances. To overcome this, we design an algorithm such that *the revision* Δr_{ij} *on the current allocation is proportional to this distance*, i.e.,

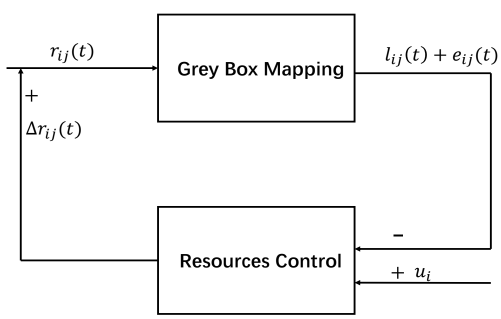


Fig. 2: Feedback Control (Grey box means that the exact form of function $l_{ij}(r_{ij})$ is unknown, but the l_{ij} value for a given r_{ij} can be estimated with a certain estimation error e_{ij})

$$r_{ij}(t{+}1) = r_{ij}(t) + \underbrace{\delta_{ij}\left(u_i - l_{ij}\left(r_{ij}\left(t\right)\right)\right)}_{\textbf{revision } \Delta \mathbf{r_{ij}}(t)}, \tag{1}$$

where $\delta_{ij} > 0$ is the proportional coefficient or stepsize.

***Remark** 1:* In this paper, we consider dynamic resource allocation that incrementally revises the current allocation based on the current performance. The term "proportional control" refers to controlling this revision in proportion to the distance between the current and target performances.

We first consider the case with accurate l_{ij} value, i.e., there is no estimation error.

***Theorem** 1:* Under Assumption 1, the performance l_{ij} converges to the target u_i if the stepsize $\delta_{ij} < \frac{2}{\mu}$.

Proof: Consider Lyapunov function $V_{ij} = \frac{1}{2}(u_i - l_{ij}(r_{ij}))^2$. By the first-order Taylor expansion,

$$\begin{aligned}
&V_{ij}(r_{ij}(t{+}1)) - V_{ij}(r_{ij}(t))\\
&= \frac{1}{2}\left(2u_i - l_{ij}(r_{ij}(t{+}1)) - l_{ij}(r_{ij}(t))\right)\\
&\quad \times \left(l_{ij}(r_{ij}(t)) - l_{ij}(r_{ij}(t{+}1))\right)\\
&= \frac{1}{2}\left(2u_i - 2l_{ij}(r_{ij}(t)) - l'_{ij}(\tilde{r}_{ij})\delta_{ij}(u_i - l_{ij}(r_{ij}(t)))\right)\\
&\quad \times \left(-l'_{ij}(\tilde{r}_{ij})\delta_{ij}(u_i - l_{ij}(r_{ij}(t)))\right)\\
&= -\frac{1}{2}\left(2 - l'_{ij}(\tilde{r}_{ij})\delta_{ij}\right) l'_{ij}(\tilde{r}_{ij})\delta_{ij}(u_i - l_{ij}(r_{ij}(t)))^2,
\end{aligned}$$

where $\tilde{r}_{ij} \in \{\alpha r_{ij}(t) + (1-\alpha) r_{ij}(t{+}1) \mid \alpha \in [0,1]\}$. If $\delta_{ij} < \frac{2}{\mu}$, $V_{ij}(r_{ij}(t{+}1)) - V_{ij}(r_{ij}(t)) \leq 0$. Further, $V_{ij}(r_{ij}(t{+}1)) - V_{ij}(r_{ij}(t)) = 0$ iff $u_i = l_{ij}(t_{ij})$. The result follows from the Lyapunov theorem [11]. ■

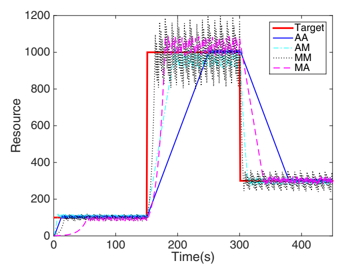


Fig. 3: Dream's Resource Allocation [1]

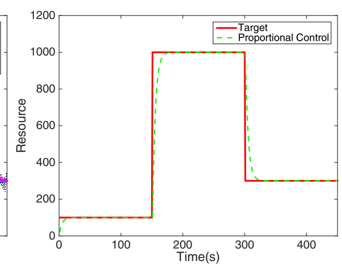


Fig. 4: Proportional Control Based Resource Allocation

Theorem 1 implies that the resource allocation algorithm (1) solves the following optimization problem:

$$\min_{r_{ij}} \quad \frac{1}{2}(u_i - l_{ij}(r_{ij}))^2. \tag{2}$$

When l_{ij} is strictly monotone, the above problem has a unique global optimum even if it may not be convex. If the exact form of function $l_{ij}(r_{ij})$ is known, problem (2) is trivial to solve. However, the exact form of $l_{ij}(r_{ij})$ is usually unknown. Instead, the l_{ij} value can be estimated for a given r_{ij}, which is why we use feedback control (1) to "learn" the optimum.

We now consider the case with estimation error in the l_{ij} value, where the resource allocation algorithm is given by

$$r_{ij}(t{+}1) = r_{ij}(t) + \delta_{ij}(u_i - l_{ij}(r_{ij}(t)) - e_{ij}(t)) \tag{3}$$

with e_{ij} the estimation error. We show that if the estimation error is bounded, under certain conditions, the algorithm (3) will converge to within an ϵ-neighborhood of the optimum, where the size of the neighborhood depends on the accuracy of the estimation. The proof is similar to that in [7].

***Theorem** 2:* Let r^*_{ij} be the fixed point of the algorithm (1). Then, there exists a $\sigma > 0$ and a unique continuously differential function ψ defined on a σ-neighborhood of the origin $e_{ij}=0$ such that $u_i - l_{ij}(\psi(e_{ij})) - e_{ij} = 0$.

Proof: At the fixed point r^*_{ij}, we have $u_i - l_{ij}(r^*_{ij})) = 0$. Since l_{ij} is strictly increasing, we have $l'_{ij} \neq 0$. The result follows directly from the implicit function theorem [10]. ∎

Theorem 3: Under Assumptions 1 and 2, if the stepsize $\delta_{ij} < \frac{2\gamma}{\mu^2+\kappa\rho}$, then for any $\epsilon > 0$, there exists a $\varepsilon > 0$ such that if $|e_{ij}| < \varepsilon$, the algorithm (3) will converge to within an ϵ-neighborhood of r^*_{ij}.

Proof: By Theorem 2, for any ϵ, there exists a ζ such that $|e_{ij}| < \zeta$ implies that $|\psi(e_{ij}) - r^*_{ij}| < \epsilon - 2\delta_{ij}\zeta$. By second-order Taylor expansion and $V''_{ij} \leq \mu^2+\kappa\rho$, we have

$$
\begin{aligned}
& V_{ij}(r_{ij}(t+1)) - V_{ij}(r_{ij}(t)) \\
&= V'_{ij}(r_{ij}(t))(r_{ij}(t+1) - r_{ij}(t)) + \frac{1}{2}V''_{ij}(\tilde{r}_{ij})(r_{ij}(t+1) - r_{ij}(t))^2 \\
&\leq (\frac{\mu^2+\kappa\rho}{2} - \frac{l'_{ij}}{\delta_{ij}})(r_{ij}(t+1) - r_{ij}(t))^2 - e_{ij}(t)l'_{ij}(r_{ij}(t+1) - r_{ij}(t)) \\
&\leq l'_{ij}|r_{ij}(t+1) - r_{ij}(t)|(e_{ij} - (\frac{1}{\delta_{ij}} - \frac{\mu^2+\kappa\rho}{2l'_{ij}})|r_{ij}(t+1) - r_{ij}(t)|\,).
\end{aligned}
$$

If $|r_{ij}(t+1) - r_{ij}(t)| \geq \varepsilon/\left(\frac{1}{\delta_{ij}} - \frac{\mu^2+\kappa\rho}{2\gamma}\right)$, V_{ij} will keep decreasing. Therefore, the algorithm (3) will enter at lease once into a region defined by $|u_i - l_{ij}(r_{ij}(t))| \leq \frac{\varepsilon}{\delta_{ij}}/\left(\frac{1}{\delta_{ij}} - \frac{\mu^2+\kappa\rho}{2\gamma}\right)$. Take any $\varepsilon < \zeta\delta_{ij}/\left(\frac{1}{\delta_{ij}} - \frac{\mu^2+\kappa\rho}{2\gamma}\right)$, and we have $|u_i - l_{ij}(r_{ij}(t))| \leq \zeta$, i.e., there exists a $\bar{e}_{ij}$ such that $|\bar{e}_{ij}| \leq \zeta$ and $u_i - l_{ij}(r_{ij}) - \bar{e}_{ij} = 0$. It follows that the algorithm will enter at least once into a neighborhood of r^*_{ij} defined by $|r_{ij} - r^*_{ij}| = |\psi(\bar{e}_{ij}) - r^*_{ij}| < \epsilon - 2\delta_{ij}\zeta$. Once it enters this neighborhood, the algorithm will stay inside a larger neighborhood defined by $\|r_{ij} - r^*_{ij}\|_2 < \epsilon$. ∎

Theorem 3 is the robust quantification of proportional control based resource allocation algorithm (1) to the estimation error.

IV. Inter-task Coupling

In this section we consider the inter-task coupling where a node j has a set $\mathcal{T}_j$ of tasks running on it. At each time, some tasks may increase the resource allocated, while others may release some resource. The resource may be exhausted at t temporarily, and in this case the increased resource at the tasks has to be upper bounded by what is released from other tasks. Denote by $\Delta^+_j(t)$ the total amount of increased allocation if there is no constraint on available resource, $\Delta^-_j(t)$ total amount of resource released and $Q_j(t)$ the amount of resource that has not been allocated yet at time t. If $\Delta^+_j(t) > \Delta^-_j(t) + Q_j(t)$, we need to revise the increased allocation to match the available resource, which will couple the resource allocation of all tasks even if their resource requirements can all be met in the end. A reasonable revision mechanism is to allocate $\Delta^-_j(t) + Q_j(t)$ to the tasks in proportion to their increased allocations when there is no constraint on available resource. For example, if task A wants 5 more units of resource and B wants 10 more units, we will allocate A 1 unit and B 2 units when the total available resource is only 3 units. Denote by Ξ^+_j the set of tasks that have increased allocation and Ξ^-_j the set of tasks that will release some resource. We have the following resource allocation algorithm:

$$
r_{ij}(t+1) = r_{ij}(t) + \min\{\frac{\Delta^-_j(t) + Q_j(t)}{\Delta^+_j(t)}, 1\}\delta_{ij}(u_i - l_{ij}(r_{ij}(t))), \quad \forall i \in \Xi^+_j(t), \tag{4}
$$

$$
r_{ij}(t+1) = r_{ij}(t) + \delta_{ij}(u_i - l_{ij}(r_{ij}(t))), \quad \forall i \in \Xi^-_j(t), \tag{5}
$$

$$
\Delta^+_j(t) = \sum_{i\in\mathcal{T}_j} \max\{u_i - l_{ij}(r_{ij}(t)), 0\}\delta_{ij} \tag{6}
$$

$$
\Delta^-_j(t) = \sum_{i\in\mathcal{T}_j} -\min\{u_i - l_{ij}(r_{ij}(t)), 0\}\delta_{ij} \tag{7}
$$

$$
Q_j(t) = Q_j(0) - \sum_{i\in\mathcal{T}_j} r_{ij}(t). \tag{8}
$$

We can show that the above algorithm is a proportional control with projection onto the set of feasible allocations under the l_1-norm.

Theorem 4: Denote by R^j, U^j and L^j the corresponding column vectors of r_{ij}, u_i and l_{ij}, $\forall i \in \mathcal{T}_j$. The algorithm (4)-(8) can be written as

$$
R^j(t+1) = \left[R^j(t) + \mathbf{diag}(\delta_{ij})(U^j - L^j(R^j(t))\right]^{+^1}_{\Omega_j}, \tag{9}
$$

where '$+^1$' denotes the projection under the l_1-norm and Ω_j is the set of feasible resource allocations at node j.

Before proving the above result, we introduce some lemmas that will be used.

Lemma 1: Let r, δ and x be n-dimensional column vector with $r + \delta \succeq \mathbf{0}$. Consider convex set $\Omega =: \{x|\mathbf{1}^T x \leq Q, x \succeq \mathbf{0}\}$ with Q a certain positive constant. Assume $r \in \Omega$, then the optimal value of the projection problem onto Ω under the l_1-norm: $\min_{y\in\Omega}\|y - (r+\delta)\|_1$ is $\max\{\mathbf{1}^T(r+\delta) - Q, 0\}$ and the minimizer may not be unique.

Proof: Let $z = r + \delta$. Then, $z \in \Omega' =: \{z|\mathbf{1}^T z = \mathbf{1}^T(r+\delta), z \succeq \mathbf{0}\}$. Ω's boundary and Ω' are parallel, with $\max\{\mathbf{1}^T(r+\delta) - Q, 0\}$ the l_1-norm distance between those two sets. So, the optimal value of the projection problem is $\max\{\mathbf{1}^T(r+\delta) - Q, 0\}$.

The l_1-norm is not a strictly convex function, so the projection problem may have multiple solutions. For example, in two-dimensional case, consider $x_1 + x_2 \leq 2, x_1 \geq 0, x_2 \geq 0$, with $x + \delta = (1.5, 1.5)$, then solution set is $\{q|q_1 + q_2 = 2, 0 \leq q_1 \leq 1.5, 0 \leq q_2 \leq 1.5\}$, which is not a singleton. ∎

Lemma 2: Without loss of generality, re-order the components of δ in Lemma 1 such that the negative components are before the positive ones, i.e., $\delta_i < 0$ for $i \leq j$ and $\delta_i > 0$ for $i > j$ where j is the number of negative components in δ_i. Consider the case $\sum_{i=1}^{j}|\delta_i| + Q - \mathbf{1}^T r \leq \sum_{i=j+1}^{n}\delta_i$. Then, the following operation

$$
y_i = r_i + \delta_i, \qquad 1 \leq i \leq j, \tag{10}
$$

$$
y_i = r_i + \frac{\delta_i}{\sum_{k=j+1}^{n}\delta_k}\left(\sum_{p=1}^{j}|\delta_p| + Q - \mathbf{1}^T r\right), j+1 \leq i \leq n \tag{11}
$$

is projection of $r+\delta$ onto the set Ω under the l_1-norm.

Proof: Consider the l_1-norm distance between y and $r+\delta$:

$$\begin{aligned}\|y-(r+\delta)\|_1 &= \sum_{i=j+1}^{n}\left(\delta_i - \frac{\delta_i}{\sum_{k=j+1}^{n}\delta_k}\big(\sum_{p=1}^{j}|\delta_p|+Q-\mathbf{1}^T r\big)\right)\\ &= \sum_{i=j+1}^{n}\delta_i - \left(\sum_{p=1}^{j}|\delta_p|+Q-\mathbf{1}^T r\right)\\ &= \mathbf{1}^T(r+\delta)-Q\end{aligned}$$

Also, $1^T y = Q, y \succeq 0$. According to Lemma 1, the operation (10)-(11) is a projection of $r+\delta$ onto the set Ω under the l_1-norm. ■

When $\sum_{i=1}^{j}|\delta_i|+Q-\mathbf{1}^T r \geq \sum_{i=j+1}^{n}\delta_i$, it means $r+\delta\in\Omega$. The optimal value is 0. So both cases are projection of $r+\delta$ onto the set Ω under the l_1-norm.

Lemmas 1 and 2 imply that when the available resource is enough or not, the allocation in proportion to the desired increase is a projection under the l_1-norm. Theorem 4 is a direct result from Lemmas 1 and 2.

As in the case under the l_2-norm [8], the projection under the l_1-norm satisfies the projection theorem:

$$(R^j(t)-R^j(t+1))^T(R^j(t)+\mathbf{diag}(\delta_{ij})(U^j-L^j(R^j(t))) -R^j(t+1))\leq 0. \tag{12}$$

***Theorem** 5:* Under Assumptions 1 and 2, the algorithm (9) converges to the target performance if the stepsize $\delta_{ij} < \frac{2\gamma}{\mu^2+\kappa\rho}$.

Proof: Consider the Lyapunov function as $V_j(R^j) = \sum_{i\in\mathcal{T}_j}\frac{1}{2}(u_i-l_{ij}(r_{ij}))^2$. By the second-order Taylor expansion,

$$\begin{aligned}&V_j(R^j(t+1))-V_j(R^j(t))\\ &= \nabla V_j(R^j(t))(R^j(t+1)-R^j(t))\\ &\quad+\frac{1}{2}(R^j(t+1)-R^j(t))^T\nabla^2 V_j(\tilde{R}_i^j)(R^j(t+1)-R^j(t))\\ &\leq \nabla V_j(R^j(t))(R^j(t+1)-R^j(t))\\ &\quad+\frac{1}{2}(\mu^2+\kappa\rho)\|R^j(t+1)-R^j(t)\|_2^2.\end{aligned}$$

By equation (12), we have $\nabla V_j(R^j(t))(R^j(t+1)-R^j(t))\leq -\frac{\gamma}{\max_i\{\delta_{ij}\}}\|r(t+1)-r(t)\|_2^2$. Thus,

$$\begin{aligned}&V_j(R^j(t+1))-V_j(R^j(t))\\ &\leq (\frac{1}{2}(\mu^2+\kappa\rho)-\frac{\gamma}{\max_i\{\delta_{ij}\}})\|r(t+1)-r(t)\|_2^2\\ &\leq 0,\end{aligned}$$

if the stepsize $\delta_{ij} < \frac{2\gamma}{\mu^2+\kappa\rho}$. The result follows from the Lyapunov theorem [11]. ■

V. Spatial Coupling

In this section we consider the spatial coupling where a task i is allocated resource from a set $\mathcal{S}_i$ of nodes to meet a target performance. The task achieves a local performance l_{ij} at node $j\in\mathcal{S}_i$, while the goal is to achieve a target global performance $G_i = u_i$. So, we need to coordinate subtasks within $\mathcal{S}_i$. Assume $G_i(t) = \sum_{j\in\mathcal{S}_i}\omega_{ij}l_{ij}(t)$ with $\sum_{j\in\mathcal{S}_i}\omega_{ij} = 1$ and $\omega_{ij} > 0$. We further assume that ω_{ij} is lower bounded by $\underline{\omega} > 0$. The resource allocation algorithm works as follows: at each time t, 1) $l_{ij}(t)$ is estimated at each node and then sent to the controller; 2) the controller computes $G_i(t)$ based on the received $l_{ij}(t)$ according to

$$G_i(t) = \sum_{j\in\mathcal{S}_i}\omega_{ij}l_{ij}(r_{ij}(t)), \tag{13}$$

and sends it to node $j\in\mathcal{S}_i$; 3) the resource allocation is updated at each node according to

$$r_{ij}(t+1) = r_{ij}(t)+\delta_{ij}(u_i-G_i(t)). \tag{14}$$

We can conclude that even we just choose G_i to coordinate all the subtasks, the global performance will converge to u_i.

***Theorem** 6:* Under the same assumptions and stepsize as in Theorem 1, the global performance converges to the target under the resource allocation algorithm (13)-(14).

Proof: The proof is similar to that of Theorem 1. We omit the detail here. ■

With both inter-task and spatial coumpings, the proportional control based resource allocation algorithm can be written as

$$R^j(t+1) = \left[R^j(t)+\mathbf{diag}(\delta_{ij})(U^j-G^j(t)\right]_{\Omega_j}^{+^1}, \tag{15}$$

where G^j is the column vector of tasks' global performance within node j. We will prove its convergence in a more general setting with asynchronous updates in the next section.

VI. Asynchronous Dynamics

In the previous sections, we have implicitly assumed that updates at different nodes and the controller are synchronous. However, in real systems, due to communication delay and packet loss, the nodes and controller may update asynchronously. In this section, we consider asynchronous resource allocation algorithm and study its convergence.

Let $T_{ij}^S \subseteq \{1,2,\cdots\}$ be the set of times at which r_{ij} is updated. At time $t\notin T_{ij}^S$, r_{ij} is unchanged, i.e., $r_{ij}(t+1) = r_{ij}(t), t\notin T_{ij}^S$. Similarly, let $T_i^C\subseteq\{1,2,\cdots\}$ be a set of times at which controller updates task $i's$ global performance. At time $t\notin T_i^C$, $G_i(t+1) = G_i(t)$. The asynchronous resource allocation works as follows:

- At time $t\in T_{ij}^S$, node j updates r_{ij} for task i based on the current G_i according to $r_{ij}(t+1) = [r_{ij}(t) + \delta_{ij}(u_i-\hat{G}_i(t))]_{\Omega_j}^{+^1}$, with $\hat{G}_i(t) = \sum_{\tau=t-t'}^{t}\beta_{ij}(\tau,t)G_i(\tau)$ and $\sum_{\tau=t-t'}^{t}\beta_{ij}(\tau,t) = 1$, where $\tau\in\{t-t',\cdots,t\}$, then sends l_{ij} to the controller.
- At times $t\in T_i^C$, the controller computes G_i for task i with the received l_{ij} according to $G_i(t) =$

$\sum_{j\in\mathcal{S}_i}\omega_{ij}\hat{l}_{ij}(t)$, with $\hat{l}_{ij}(t)=\sum_{\tau=t-t'}^{t}\eta_{ij}(\tau,t)l_{ij}(\tau)$ and $\sum_{\tau=t-t'}^{t}\eta_{ij}(\tau,t)=1$, then sends G_i to node $j\in\mathcal{S}_i$.

***Remark** 2:* In general, asynchronous model has two common types of policies [9]. One type uses the latest average information for updates, which is what we adopt. The other uses the latest information for updates. For this one, we can just set $\beta_{ij}(\tau,t)$ and $\eta_{ij}(\tau,t)$ to be 1 for $\tau=\max\{t-t',\cdots,t\}$, and set the values to 0 otherwise. So, these two types are both included in our asynchronous resource allocation algorithm.

***Assumption** 3:* For all updates on different nodes and the controller, the time t' between consecutive updates is bounded, e.g., $t'\le\Gamma$.

Let $r_{ij}(t)$ be the "ideal" resource amount if node j knows the exact $G_i(t)$ at time t, and $\hat{r}_{ij}$ the amount in the asynchronous algorithm. $G_i(t)$ and $\hat{G}_i(t)$ are defined similarly. Let $\pi_{ij}(t)=r_{ij}(t+1)-r_{ij}(t)$ and π_i denote the column vector of π_{ij}. It is straightforward to verify that $\gamma\le\nabla_{r_{ij}}G_i\le\mu$.

***Lemma** 3:* $|\hat{G}_i(t)-G_i(t)|\le\mu\sum_{\tau=t-t'}^{t-1}\|\pi_i(\tau)\|_1$.

Proof: We have

$$
\begin{aligned}
&\left|\hat{G}_i(t)-G_i(t)\right|\\
&=\left|\sum_{j\in T_i}\omega_{ij}\left(\hat{l}_{ij}(t)-l_{ij}(t)\right)\right|\\
&=\left|\sum_{j\in T_i}\omega_{ij}\left(\sum_{\tau=t-t'}^{t}\eta_{ij}(\tau,t)l_{ij}(\tau)-l_{ij}(t)\right)\right|\\
&\le\sum_{j\in T_i}\omega_{ij}\left(\max_{t-t'\le\tau'\le t}|l_{ij}(\tau')-l_{ij}(t)|\right)\\
&\le\sum_{j\in T_i}\omega_{ij}\left(\max_{t-t'\le\tau'\le t}\mu\,|r_{ij}(\tau')-r_{ij}(t)|\right)\\
&\le\sum_{j\in T_i}\omega_{ij}\left(\max_{t-t'\le\tau'\le t}\mu\sum_{\tau=\tau'}^{t-1}|\pi_{ij}(\tau)|\right)\\
&\le\mu\sum_{j\in T_i}\omega_{ij}\sum_{\tau=t-t'}^{t-1}|\pi_{ij}(\tau)|\\
&\le\mu\sum_{\tau=t-t'}^{t-1}\|\pi_i(\tau)\|_1,
\end{aligned}
$$

where the second inequality follows from the mean value theorem. ■

***Theorem** 7:* Under Assumptions 1, 2 and 3, if the stepsize satisfies $\delta_{ij}<\underline{\omega}\gamma/\left(\frac{\mu^2+\kappa\rho}{2}+2(\Gamma+1)\mu^2\right)$, the difference $|\hat{G}_i(t)-G_i(t)|$ due to asynchronous updates approaches to zero as $t\to\infty$.

Proof: Denote by R_i the column vector of $r_{ij}\ \forall j\in\mathcal{S}_i$. Let $q(t)\in\{\ \vartheta|\vartheta=\alpha R_i(t)+(1-\alpha)R_i(t+1)\ |\ \alpha\in[0,1]\}$. Define Lyapunov function as $V_i=\sum_{j\in\mathcal{S}_i}\frac{1}{2}(u_i-l_{ij}(r_{ij}))^2$, then we have:

$$
\begin{aligned}
&V_i(t+1)-V_i(t)\\
&=\underbrace{\nabla V_i(t)^T\pi_i(t)}_{\textbf{(a)}}+\underbrace{\frac{1}{2}\pi_i(t)^T\nabla^2V_i(q(t))\pi_i(t)}_{\textbf{(b)}}.
\end{aligned}
$$

Consider the part **(a)**:

$$
\begin{aligned}
\textbf{(a)}&=\Big(\nabla_{r_i}G_i(u_i-\hat{G}_i(t))-\nabla_{r_i}G_i(u_i-G_i(t))\Big)^T\pi_i(t)\\
&\quad-\nabla_{r_i}G_i(u_i-\hat{G}_i(t))\pi_i(t)\\
&=\nabla_{r_i}G_i(G_i(t)-\hat{G}_i(t))\pi_i(t)-\nabla_{r_i}G_i(u_i-\hat{G}_i(t))\pi_i(t)\\
&\le\mu\left|G_i(t)-\hat{G}_i(t)\right|\|\pi_i(t)\|_1-\nabla_{r_i}G_i(u_i-\hat{G}_i(t))\pi_i(t)\\
&\le\mu^2\sum_{\tau=t-t'}^{t-1}\|\pi_i(\tau)\|_1\|\pi_i(t)\|_1-\nabla_{r_i}G_i(u_i-\hat{G}_i(t))\pi_i(t)\\
&\le\mu^2\sum_{\tau=t-t'}^{t}\|\pi_i(\tau)\|_2^2\underbrace{-\nabla_{r_i}G_i(u_i-\hat{G}_i(t))\pi_i(t)}_{\textbf{(c)}}.
\end{aligned}\tag{16}
$$

The second inequality follows from Lemma 3. Considering the part **(c)**, we find that (15) satisfies the projection theorem even if the tasks are not within one node since our special operation based on l_1-norm has following properties:

$$
\begin{aligned}
(r_{ij}(t)-r_{ij}(t+1))(r_{ij}(t)+\delta_{ij}(u_i\\
-G_i(t))-r_{ij}(t+1))\le 0,
\end{aligned}\tag{17}
$$

$$
(u_i-G_i(t))(r_{ij}(t+1)-r_{ij}(t))\ge 0.\tag{18}
$$

With (17) and (18), we apply the same method in the part **(c)** and obtain:

$$
\begin{aligned}
\textbf{(c)}&=\nabla_{r_i}G_i(\hat{G}_i(t)-G_i(t))\pi_i(t)-\nabla_{r_i}G_i(u_i-G_i(t))\pi_i(t)\\
&\le\mu^2\sum_{\tau=t-t'}^{t}\|\pi_i(\tau)\|_2^2-\nabla_{r_i}G_i(u_i-G_i(t))\pi_i(t)\\
&\le\mu^2\sum_{\tau=t-t'}^{t}\|\pi_i(\tau)\|_2^2+\sum_{j\in T_i}-\frac{\omega_{ij}\gamma}{\delta_{ij}}\pi_{ij}(t)^2\\
&\le\mu^2\sum_{\tau=t-t'}^{t}\|\pi_i(\tau)\|_2^2-\frac{\underline{\omega}\gamma}{\max\{\delta_{ij}\}}\|\pi_i(t)\|_2^2.
\end{aligned}\tag{19}
$$

Combining (16) and (19), we can get the final form of **(a)**:

$$
\textbf{(a)}\le 2\mu^2\sum_{\tau=t-t'}^{t}\|\pi_i(\tau)\|_2^2-\frac{\underline{\omega}\gamma}{\max\{\delta_{ij}\}}\|\pi_i(t)\|_2^2.
$$

With part **(a)** and **(b)**, we have:

$$
\begin{aligned}
V_i(t+1)-V_i(t)&\le 2\mu^2\sum_{\tau=t-t'}^{t}\|\pi_i(\tau)\|_2^2-\frac{\underline{\omega}\gamma\|\pi_i(t)\|_2^2}{\max\{\delta_{ij}\}}\\
&\quad+\frac{\mu^2+\kappa\rho}{2}\|\pi_i(t)\|_2^2.
\end{aligned}\tag{20}
$$

Adding up equation (20) over all t, we have the following result:

$$
V_i(t+1)-V_i(0)
$$

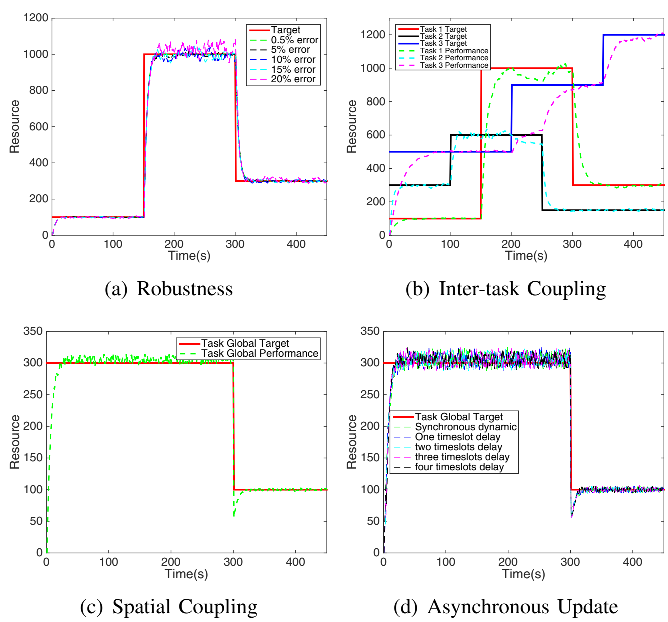
$$
\begin{aligned}
&\leq \left(\frac{\mu^2 + \kappa\rho}{2} - \frac{\omega\gamma}{\max\{\delta_{ij}\}} \right) \sum_{\tau=0}^{t} \|\pi_i\|_2^2 \\
&\quad + 2\mu^2 \sum_{\tau'=0}^{t} \sum_{\tau=\tau'-t'}^{\tau_\prime} \|\pi_i(\tau)\|_2^2 \\
&\leq - \left(\frac{\omega\gamma}{\max\{\delta_{ij}\}} - \frac{\mu^2 + \kappa\rho}{2} - 2(t'+1)\mu^2 \right) \sum_{\tau=0}^{t} \|\pi_i(\tau)\|_2^2.
\end{aligned}
$$

Under Assumption 3, with V_i bounded and stepsize constraint, when $t \to \infty$, we can obtain the result that $\sum_{t=0}^{\infty} \|\pi_i(t)\|_2^2 < \infty$. So we must have $\|\pi_i(t)\|_2 \to 0$ with $t \to \infty$. By Lemma 3, we have $|\hat{G}_i(t) - G_i(t)| \leq \mu \sum_{j \in T(i)} \sum_{\tau=t-t'}^{t-1} |\pi_{ij}(\tau)|$. Based on this, we get $\lim_{t\to\infty} |\hat{G}_i(t) - G_i(t)| = 0$. ■

Also, notice that based on equation (20), it is straightforward to establish the convergence of the asynchronous resource allocation algorithm with properly chosen stepsize.

VII. Numerical Examples

Numerical examples have confirmed our theoretical results. In numerical experiments, the target performance is not fixed, and we use linear and quadratic functions for l_{ij}. At the beginning, all the tasks have zero resource. We evaluate our resource allocation algorithm in terms of robustness to estimation error, inter-task coupling, spatial coupling and asynchronous update.



(a) Robustness

(b) Inter-task Coupling

(c) Spatial Coupling

(d) Asynchronous Update

Fig. 5: Numerical Examples

Fig. 5(a) shows the robustness to estimation error. We experiment with error ranges of $0.5\%, 5\%, 10\%, 15\%, 20\%$. We see that even with 20% estimation error, our algorithm converges to within a small neighborhood of the optimal allocation.

In Fig. 5(b), we experiment with inter-task coupling with a 15% estimation error for each task. There are 3 tasks at the node. The resource capacity is 2100 units which is not enough temporally to accommodate all tasks' desired allocations at all time but is sufficient to finally meet all tasks' requirements. For example, at the timeslot 250, the resource is not enough temporarily where our algorithm will do the projection based on l_1-norm. However, all the tasks' resource requirements are met in the end.

In Fig. 5(c), we consider a situation when a task consists of 3 different subtasks at three different nodes with 2.5% estimation error. At the timeslot $1-300$, subtask 1 needs 300 units, subtask 2 needs 250, subtask 3 needs 350. At time slot $301-450$, all subtasks need 100 units. We see that the system achieves the allocation and performance.

In Fig. 5(d), we evaluate resource allocation under the asynchronous update. The updates are delayed by 1, 2, 4 and 4 timeslots. Also, each task is with 5% estimation error and its allocation is updated based on the average global performance. As we can see, even with 4-timeslot delay, the distance between synchronous and asynchronous updates becomes small enough as the time goes by.

VIII. Conclusion

In this paper, we have shown that for a very general resource allocation problem, algorithms based on the proportional control achieve both fast response and convergence. We have established the convergence of the proportional control algorithms with inter-task and spatial couplings, quantified their performance under estimation error, and established the convergence of the algorithms when operating asynchronously with the bounded delay. Numerical experiments are provided to complement the theoretical analysis.

Acknowledgement

The authors would like to thank Guohui Ding and Xinyang Zhou for helpful discussion.

References

[1] Moshref, M., Yu, M., Govindan, R. and Vahdat, A., DREAM: dynamic resource allocation for software-defined measurement. ACM SIGCOMM Computer Communication Review, 44(4): 419–430, 2015.

[2] Claise, B., Cisco systems NetFlow services export version 9, 2004.

[3] Wang, M., Li, B. and Li, Z., sFlow: Towards resource-efficient and agile service federation in service overlay networks. In Distributed Computing Systems, Proceedings. 24th International Conference on, 628–635, 2004.

[4] Moshref, M., Yu, M. and Govindan, R., Resource/accuracy tradeoffs in software-defined measurement. In Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking, 73–78, 2013.

[5] Curtis, A., Mogul, J., Tourrilhes, J., Yalagandula, P., Sharma, P. and Banerjee, S., DevoFlow: Scaling Flow Management for High-Performance Networks. In SIGCOMM, 2011.

[6] O'Dwyer, A., Handbook of PI and PID controller tuning rules. 2009.

[7] Chen, L., Low, S.H. and Doyle, J.C., Random access game and medium access control design. IEEE/ACM Transactions on Networking, 18(4):1303–1316, 2010.

[8] Bertsekas, D.P. and Tsitsiklis, J.N., Parallel and distributed computation: numerical methods. 1989.

[9] Low, S.H. and Lapsley, D.E., Optimization flow controlI: basic algorithm and convergence. IEEE/ACM Transactions on Networking, 7(6): 861–874, 1989.

[10] Abraham, R., Marsden, J.E. and Ratiu, T., Manifolds, tensor analysis, and applications. 2012.

[11] Khalil, H.K., Noninear Systems. Prentice-Hall, New Jersey, 1996.