An Aggregation Approach for Solving the Non-linear Fractional Equality Knapsack Problem

Anis Yazidi[a], Tore M. Jonassen[a], Enrique Herrera-Viedma[b,c]

[a] *Department of Computer Science,*
Oslo and Akershus University College of Applied Sciences,
Oslo, Norway.
[b] *Department of Computer Science and Artificial Intelligence,*
University of Granada,
Granada, Spain.
[c] *Department of Electrical and Computer Engineering,*
Faculty of Engineering, King Abdulaziz University,
Jeddah 21589, Saudi Arabia.

Abstract

In this paper, we present an *optimal*, efficient and yet simple solution to a class of the *deterministic* non-linear fractional equality knapsack (NEFK) problem — a substantial resource allocation problem. The solution is shown to be superior to the state-of-the-art in terms of convergence speed.

We provide a rigorous analysis that proves the optimality of our scheme under general conditions. Our solution resorts to a subtle aggregation procedure that drives the system towards equalizing the derivatives of the material value functions in a similar manner to the Homo Egualis theory. Furthermore, we report experimental results that catalogue the applicability of our solution to the problem of rate limiting in cloud computing, which falls under the *deterministic* NEFK problem.

Keywords:
Non-Linear Fractional Equality Knapsack, Resource Allocation, Dynamical System, Rate Limiting.

1. Introduction

The self-optimization aspect of Autonomic Computing (AC) systems (Kephart and Chess, 2003) envisages dynamic allocation of a shared resource pool between applications in order to yield optimal resource usage. In more formal terms, the allocation is viewed as a vector p, where each component p_i represents the share of resources of a consumer i (Loureiro et al., 2012). Thus, the problem reduces to an optimization problem with constraint on the resource capacity. The aim is

Email addresses: anis.yazidi@hioa.no (Anis Yazidi), toremj@gmail.com (Tore M. Jonassen), viedma@decsai.ugr.es (Enrique Herrera-Viedma)

to find an optimal allocation vector p^* that maximizes a performance function f that depends on p. We will show that the latter problem is an instance of the *deterministic* non-linear fractional equality (Granmo and Oommen, 2011; Granmo et al., 2007; Granmo and Oommen, 2010b,a). While there are myriad solutions to different classes of knapsack problems (He et al., 2016), most of them are static and are based on the worst case or average case scenarios. Unlike the realm of convex optimization, NEFK uses a monotonicity assumption, which is common in many real-life problems, including resource allocation problems. In this paper, we will show that, by appropriately defining f, we are able to deal with two distinct problems, namely: a utility optimization problem (Stanojevic and Shorten, 2010, 2009b) and a fairness problem (Stanojevic and Shorten, 2009b, 2008) with constraint on the resource capacity. In order to put our work in the right perspective, we emphasize that the *deterministic* NEFK problem (Granmo and Oommen, 2011, 2010b,a) allows for a large set of applications, including:

- A typical application is the problem of distributed rate limiting for cloud-based services investigated in Stanejevic et al. (Stanojevic and Shorten, 2008, 2009a) where the resource to be shared among traffic limiters is the bandwidth capacity. An optimal solution in this case aspires to achieve a fairness postulate that states that: "the performance levels at different servers should be (approximately) equal" (Stanojevic and Shorten, 2009a).
- Dynamic speed scaling of processes in cloud computing based on demand and performance constraints in order to minimize energy consumption (Stanojevic and Shorten, 2010).
- A class of utility maximization problems (Loureiro et al., 2012) for resource allocation.

Our model for efficiently solving the NEFK problem is based on the theory of dynamical systems. The model is simple and can be easily implemented in a centralized or decentralized manner thanks to a"dissemination" property of our devised algorithm. Our solution uses a subtle aggregation procedure to drive the system towards equalizing the derivatives of the material value functions in a similar manner to the Homo Egualis theory (De Jong et al., 2008a).

The rest of this paper is structured as follows. Section 2 provides the preliminaries needed for the rest of the paper. Section 3 describes the details of our solution as well as the main theoretical results of the paper. In Section 4, we provide some experimental results that catalogue the convergence of our approach, its conformance to the theoretical findings, and its superiority to the state of the art.

2. Background

The problem we address in this paper is referred to as a *non-linear fractional equality knapsack (NEFK) problem* (Granmo and Oommen, 2011; Granmo et al., 2007; Granmo and Oommen, 2010b,a).

Deterministic non-linear equality fractional knapsack (NEFK) problem: involves n materials p_i, $1 \leq i \leq n$, where each material p_i is available in a certain amount $0 \leq p_i \leq b_i$. Let $f_i(p_i)$ denote the value of the amount p_i of material i. The problem is to fill a knapsack of fixed volume $c > 0$ with the material mix $\vec{p} = [p_1, \ldots, p_n]$ of maximal value $\sum_1^n f_i(p_i)$ (Black, 2004).

The *nonlinear equality* FK problem is characterized by a separable objective function. The problem can be stated as follows (Kellerer et al., 2004):

$$\begin{array}{ll} \text{maximize} & f(\vec{p}) = \sum_{i=1}^n f_i(p_i) \\ \text{subject to} & \sum_{i=1}^n p_i = c \ \text{ and } \ p_i \geq 0 \text{ for } i \in \{1, \ldots, n\}. \end{array}$$

We suppose that the derivatives of the material value functions $f_i(p_i)$ with respect to p_i, (hereafter denoted f_i'), are non-increasing. In other words, the material value *per unit volume* is no longer constant as in the linear case, but decreases with the material amount, and so the optimization problem becomes:

$$\begin{array}{ll} \text{maximize} & f(\vec{p}) = \sum_{i=1}^n f_i(p_i), \ \text{where } f_i(p_i) = \int_0^{p_i} f_i'(\xi_i) d\xi_i \\ \text{subject to} & \sum_{i=1}^n p_i = c \ \text{ and } \ p_i \geq 0 \text{ for } i \in \{1, \ldots, n\}. \end{array}$$

Efficient solutions have been devised to the latter problem based on the principle of Lagrange multipliers. In short, the optimal value occurs when the derivatives f_i' of the material value functions are equal, subject to the knapsack constraints (Bretthauer and Shetty, 2002):

$$\begin{array}{l} f_1'(p_1) = \cdots = f_n'(p_n) \\ \sum_{i=1}^n p_i = c \ \text{ and } \ p_i \geq 0 \text{ for } i \in \{1, \ldots, n\}. \end{array}$$

2.1. State-of-the-art for deterministic NEFK and its applications to resource allocation

Palomar and Chiang introduced the concept of decomposition for utility maximization (Palomar and Chiang, 2006). Their approach relies on a distributed algorithm, where the optimization is performed over separable functions that can be regarded as a distributed version of gradient ascent algorithm (Mosk-Aoyama et al., 2010). Other approaches resort to a form of hierarchy, where the decomposition methods used by Palomar and Chiang (Palomar and Chiang, 2006) are a representative example. The essence of these methods is to decompose an optimization problem into smaller ones, where each of them can be further decomposed in a hierarchical manner. In order to place our paper within the larger body of scientific research, we briefly present here some representative studies on resource allocation in autonomic systems. Utility is a relatively recent concept in computer science that has been borrowed from the field of economics. Utility originally describes a measure of preferences over some set of goods. Traditionally, Quality of Service (QoS) was defined in a binary manner, i.e., either met or unmet. Thus, many different allocations can be "optimal" in that framework as long as the constraints of the system in terms of QoS are satisfied. By introducing utility functions (Bennani et al., 2005; Fulp et al., 1998), an allocation fitness can be measured in a non-binary manner in order to better quantify the satisfaction of a consumer with his share of a resource.

In many real-life problems, utilities are usually related in a *monotonic* manner to the amount of allocation. This monotonic property of the utility reflects the fact that the performance of a system improves as more resources are allocated to it. For instance, when it comes to a web service, the QoS measured in terms of response time degrades monotonically as less CPU and memory resources are allocated to the web server. Menasce and his collaborators (Bennani et al., 2005) have advocated using performance-dependent utility functions in resource allocation problems in distributed computing environments. Despite the fact that the utility function is usually monotonic when it comes to resource allocation, most of the available optimization paradigms have been developed for convex functions. To the best of our knowledge, the only available work in the literature that operates under the same assumption as our proposed approach is by Stanojevic and colleagues (Stanojevic and Shorten, 2008, 2009a). However, the latter work (Stanojevic and Shorten, 2008, 2009a) requires message exchange between pairs of the components of the allocation vector. In contrast to this, our algorithm is less computationally complex since it does not assume any pairwise message exchange. We will provide some experimental results that show the superiority of our scheme to the approach of Stanojevic et al. (Stanojevic and Shorten, 2008, 2009a).

Wang et al. (2008) employed a constrained non-linear optimization technique, combining both deterministic and stochastic optimization algorithms, to dynamically allocate server capacity with the help of analytical models. Another body of methods for resource allocation that are radically different from the approach presented in this paper include reinforcement learning (Dutreilh et al., 2011) and negotiations between distributed agents (Boutilier et al., 2002). In the realm of resource allocation problems, many combinatorial methods have been used such as simulated annealing, and genetic algorithms. However, these approaches do not fall under the scope of this article. Furthermore, it is worth mentioning that myriad nature-inspired approaches can be found in the literature (Gao and Pan, 2016), but they are also outside the scope of this article. An interesting concept in resource allocation, which emanates from the field of microeconomics, and more particularly from regulation in competitive markets, is tâtonnement (Subramoniam et al., 2002). Tâtonnement aims to balance demand and consumption using smart pricing techniques. The idea behind tâtonnement is simply to increase the price of the resource whenever the demand is less than consumption, in order to encourage consumption, while decreasing the price in the opposite case. This concept was applied in computer systems by assigning fictive budgets to the different applications competing for the shared resources.

Our devised solution in this article tries to equalize the derivatives of the material value functions. The solution is similar to the Homo Egualis theory (De Jong et al., 2008a). An agent in a Homo Egualis society compares his payoff to the payoff of the rest of the agents. Each agent will tend to increase his payoff in proportion to the payoff of agents that have a better payoff than him, and decrease his payoff in proportion to the payoff of agents that have lower payoffs than him (De Jong et al., 2008a; Xing and Chandramouli, 2008; De Jong et al., 2008b; de Jong and Tuyls, 2011). Instead, our solution equalizes the derivatives of the material value functions by comparing the derivative of the agent in question to the "mean" of the derivatives of all agents. This simple and subtle principle allows the system to converge towards an optimal solution.

The novel contributions of this paper are listed below:

- We devise an optimal and yet efficient solution to the deterministic NEFK problem under general conditions.
- We provide sound theoretical results that demonstrate the optimality of our solution based on analyzing the resulting dynamical system and resorting to a subtle *perturbation argument*. Unlike the mainstream literature, which deals with optimizing convex objective functions, the NEFK problem involves monotonic functions.
- Experimental results show that our solution is significantly superior to the state-of-the-art solution by Stanojevic (Stanojevic and Shorten, 2009b, 2008; Loureiro et al., 2012; Stanojevic and Shorten, 2009a) in terms of convergence speed. Moreover, our solution is less computationally complex since we do not require any message exchange as in Stanojevic et al. (Stanojevic and Shorten, 2009b, 2008, 2009a).

3. A New Method for Solving the deterministic NEFK problem

Our aim is to find a scheme that moves towards optimizing the following NEFK problem online:

$$\begin{array}{ll} \text{maximize} & f(\vec{p}) = \sum_{i=1}^{n} f_i(p_i), \text{where } f_i(p_i) = \int_0^{p_i} d_i(\xi) d\xi \\ & \text{and } d_i(p_i) = f_i'(p_i). \\ \text{subject to} & \sum_{i=1}^{n} p_i = c \text{ and } p_i \geq 0 \;\; \text{for} \;\; i \in \{1, \ldots, n\}. \end{array}$$

We shall characterize the optimal solution to the identified deterministic NEFK problem (Granmo and Oommen, 2011, 2010b,a).

Theorem 1. *The material mix $\vec{p} = [p_1, \ldots, p_n]$ is a solution to a given NEFK if (1) the derivatives of the expected material amount values are all equal at $\vec{p}$, (2) the mix fills the knapsack, and (3) every material amount is positive, i.e.:*

$$\begin{array}{l} f_1'(p_1) = \cdots = f_n'(p_n) \\ \sum_1^n p_i = c \text{ and } \forall i \in \{1, \ldots, n\}, p_i \geq 0. \end{array}$$

The above theorem is based on the well-known principle of Lagrange Multipliers, and its proof is therefore omitted here for the sake of brevity.

At this juncture, we will provide an algorithm for solving the deterministic NEFK problem.

3.1. Algorithm for solving the deterministic NEFK problem

Consider a system F in the form of $F(P(t)) = P(t+1) = P(t) - \theta D(P(t))$, where $P = (p_1, p_2, \dots, p_n)$, $t \in \mathbb{N}$ and $0 < \theta < 1$ as given below.

Primarily, we want F to be a system on the unit n-cube, that is

$$F : [0,1]^n \longrightarrow [0,1]^n$$

Let the functions $d_i : [0,1] \longrightarrow [0,1]$ be continuously differentiable (at least C^1) and strictly decreasing, that is $x < y$ implies $d_i(x) > d_i(y)$.

Consider $F : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ given by

$$\begin{aligned}
p_1(t+1) &= p_1(t) - \theta\left(\frac{1}{n}\sum_{i=1}^{n} d_i(p_i(t)) - d_1(p_1(t))\right)\\
p_2(t+1) &= p_2(t) - \theta\left(\frac{1}{n}\sum_{i=1}^{n} d_i(p_i(t)) - d_2(p_2(t))\right)\\
&\vdots\\
p_n(t+1) &= p_n(t) - \theta\left(\frac{1}{n}\sum_{i=1}^{n} d_i(p_i(t)) - d_n(p_n(t))\right)
\end{aligned} \tag{1}$$

We want to prove, under suitable conditions, that this system has a unique attracting fixed-point. The argument is valid for any hyperplane

$$\sum_{i=1}^{n} p_i(t) = c$$

$c > 0$. Without loss of generality, we will use $c = 1$. It turns out that these hyperplanes are invariant, and that they have an attracting fixed-point. The main observation is that we have a linear system in the $d_i(p_i)$.

If

$$\sum_{i=1}^{n} p_i(t) = 1$$

then

$$\begin{aligned}
\sum_{i=1}^{n} p_i(t+1) &= \sum_{i=1}^{n} p_i(t) - \theta\left(n \cdot \frac{1}{n}\sum_{i=1}^{n} d_i(p_i(t)) - \sum_{i=1}^{n} d_i(p_i(t))\right)\\
&= 1 - \theta\left(\sum_{i=1}^{n} d_i(p_i(t)) - \sum_{i=1}^{n} d_i(p_i(t))\right)\\
&= 1
\end{aligned}$$

Hence

$$\sum_{i=1}^{n} p_i(t) = 1 \implies \sum_{i=1}^{n} p_i(t+1) = 1$$

However, this does **not** ensure that all $p_i(t+1) \in [0,1]$. We must have

$$0 \le p_i(t) - \theta \left(\frac{1}{n} \sum_{j=1}^{n} d_j(p_j(t)) - d_i(p_i(t)) \right) \le 1$$

for $i = 1, 2, \ldots, n$.

Note that this simple calculation also shows that if

$$\sum_{i=1}^{n} p_i(t) = k$$

then

$$\sum_{i=1}^{n} p_i(t+1) = k$$

whenever this makes sense with respect to the other quantities involved in the definition of the system. This means that the dynamic of the system "lives" in hyperplanes of codimension 1, all of them parallel to $p_1 + p_2 + \cdots + p_n = 0$.

3.2. The fixed-point equation

Theorem 2. *Assume the notation above, and assume that the functions d_i are all strictly decreasing. Let*

$$\begin{aligned} M_i &= \max_{\xi \in [0,1]} d_i(\xi) = d_i(0) \\ m_i &= \min_{\xi \in [0,1]} d_i(\xi) = d_i(1) \\ I_i &= [m_i, M_i] \end{aligned}$$

and

$$\begin{aligned} m &= \max_{i \in \{1,\ldots,n-1\}} (\min_{\xi \in [0,1]} d_i(\xi)) = \max_{i \in \{1,\ldots,n-1\}} d_i(1) \\ M &= \min_{i \in \{1,\ldots,n-1\}} (\max_{\xi \in [0,1]} d_i(\xi)) = \min_{i \in \{1,\ldots,n-1\}} d_i(0) \end{aligned}$$

and assume $m < M$. Let

$$J = \bigcap_{i=1}^{n-1} I_i = [m, M]$$

and $d_n([0,1]) = I_n$. Assume $I_n \cap J \neq \varnothing$. Let

$$h(\xi) = \xi + \sum_{j=1}^{n-1} d_j^{-1} \circ d_n(\xi)$$

and

$$d_n^{-1}(I_n \cap J) = [\xi_l, \xi_r] \subset [0,1] \text{ where } \xi_l < \xi_r$$

Assume that $h(\xi_l) < 1$ and $h(\xi_r) > 1$.
Then the system (1) has a unique fixed point in $[0,1]^n$.

Proof. The fixed-point equation in vector form is given by

$$F(P) = P \text{ with } P = (p_1, \ldots, p_n) \text{ and } \sum_{i=1}^{n} p_i = 1$$

Hence we must have $D(P) = 0$. We write

$$p_n = 1 - \sum_{i=1}^{n-1} p_i$$

and in component form we then have for $j = 1, \ldots, n-1$

$$\frac{1}{n}\left(\sum_{i=1}^{n-1} d_i(p_i) + d_n(1 - \sum_{i=1}^{n-1} p_i)\right) - d_j(p_j) = 0$$

and

$$\frac{1}{n}\left(\sum_{i=1}^{n-1} d_i(p_i) + d_n(1 - \sum_{i=1}^{n-1} p_i)\right) - d_n(1 - \sum_{i=1}^{n-1} p_i) = 0$$

for $j = n$. Rearranging these, we get

$$\sum_{i=1}^{n-1} d_i(p_i) + d_n(1 - \sum_{i=1}^{n-1} p_i) + (1-n)d_j(p_j) = 0$$

for $j = 1, \ldots, n-1$, and

$$\sum_{i=1}^{n-1} d_i(p_i) + (1-n)d_n(1 - \sum_{i=1}^{n-1} p_i) = 0$$

for $j = n$. The equations give a linear system for the vector

$$v = (d_1(p_1), d_2(p_2), \ldots, d_{n-1}(p_{n-1}), d_n(1 - \sum_{i=1}^{n-1} p_i))$$

with $n \times n$-coefficient matrix C with $1-n$ along the diagonal, and the rest of the entries equal to 1.

$$C = \begin{bmatrix} 1-n & 1 & 1 & \cdots & 1 \\ 1 & 1-n & 1 & \cdots & 1 \\ \vdots & & \ddots & & \vdots \\ 1 & 1 & 1 & \cdots & 1-n \end{bmatrix}$$

Let L_j denote the j-ht row in C, then

$$L_1 = -\sum_{j=2}^{n} L_j \tag{2}$$

since each column in C consists of $n-1$ entries 1 and one entry $1-n$. Hence the rows are linearly dependent, so $\det C = 0$.

Now (2) gives

$$L_n = -\sum_{j=1}^{n-1} L_j$$

so the matrix C reduces to

$$\hat{C} = \begin{bmatrix} 1-n & 1 & 1 & \cdots & 1 \\ 1 & 1-n & 1 & \cdots & 1 \\ \vdots & & \ddots & & \vdots \\ 1 & 1 & \cdots & 1-n & 1 \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

The reduced row echelon form of $\hat{C}$ is given by

$$C_R = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & -1 \\ 0 & 1 & 0 & \cdots & 0 & -1 \\ \vdots & \ddots & \ddots & & \vdots & \vdots \\ 0 & \cdots & \cdots & 0 & 1 & -1 \\ 0 & 0 & \cdots & \cdots & 0 & 0 \end{bmatrix}$$

To see this, let L_j, $j = 1, \ldots, n-1$ be the j-th row in $\hat{C}$, and K_i, $i = 1, \ldots, n-1$, be the i-th row of C_R. Then

$$L_j = \sum_{\substack{i=1 \\ i \neq j}}^{n-1} K_i + (1-n) K_j$$

since this gives 1 in each column $i \neq j$ and $(1-n)$ in column j, $j = 1, \ldots, n-1$. In column n we get

$$(-1)(n-2) + (1-n)(-1) = 2 - n + n - 1 = 1$$

Hence the system $Cv = 0$ reduces to the equations

$$d_j(p_j) = d_n(1 - \sum_{i=1}^{n-1} p_i) = d_n(p_n) \text{ where } j = 1, \ldots, n-1$$

with the additional requirement that

$$\sum_{i=1}^{n} p_i = 1$$

The functions d_i are all strictly decreasing and hence have unique inverse functions d_i^{-1}, each defined in the range of d_i, an interval contained in $[0,1]$. Now since $d_j(p_j) = d_n(p_n)$ we obtain $d_j^{-1} \circ d_j(p_j) = d_j^{-1} \circ d_n(p_n)$, so $p_j = d_j^{-1} \circ d_n(p_n)$ provided that $d_n(p_n)$ is in the range of each d_j for $j = 1, \dots, n-1$.

Let

$$d_i : [0,1] \longrightarrow I_i \subset [0,1]$$

and

$$\begin{aligned} M_i &= \max_{\xi \in [0,1]} d_i(\xi) = d_i(0) \\ m_i &= \min_{\xi \in [0,1]} d_i(\xi) = d_i(1) \\ I_i &= [m_i, M_i] \end{aligned}$$

Let

$$\begin{aligned} m &= \max_{i \in \{1,\dots,n-1\}} (\min_{\xi \in [0,1]} d_i(\xi)) = \max_{i \in \{1,\dots,n-1\}} d_i(1) \\ M &= \min_{i \in \{1,\dots,n-1\}} (\max_{\xi \in [0,1]} d_i(\xi)) = \min_{i \in \{1,\dots,n-1\}} d_i(0) \end{aligned}$$

and assume $m < M$. Then $J \neq \varnothing$ where

$$J = \bigcap_{i=1}^{n-1} I_i = [m, M]$$

Let $d_n([0,1]) = I_n$. Then, from the above, we have two different cases

$$I_n \cap J = \varnothing \tag{3}$$

$$I_n \cap J \neq \varnothing \tag{4}$$

where (3) implies that we have no possibility of finding a solution in the required range of states, $p_i \in [0,1]$, but (4) makes solutions possible, but does not prove their existence. Note that (3), and hence (4), are easily checked.

Assume (4) holds, and consider

$$h : d_n^{-1}(I_n \cap J) \longrightarrow \mathbb{R}_0^+$$

where

$$h(\xi) = \xi + \sum_{j=1}^{n-1} d_j^{-1} \circ d_n(\xi)$$

Hence our problem in this case reduces to solving the equation

$$h(\xi) = 1 \text{ with } \xi \in d_n^{-1}(I_n \cap J) \tag{5}$$

Note that $d_n^{-1}(I_n \cap J) \neq \varnothing$ is a closed interval

$$d_n^{-1}(I_n \cap J) = [\xi_l, \xi_r] \subset [0,1] \text{ where } \xi_l < \xi_r$$

Furthermore, $h = h(\xi)$ is a strictly increasing function on $[\xi_l, \xi_r]$ as

$$h'(\xi) = 1 + \sum_{j=1}^{n-1} (d_j^{-1})'(d_n(\xi)) d_n'(\xi)$$

where $(d_j^{-1})'(d_n(\xi)) < 0$ and $d_n'(\xi) < 0$, so the sum of composite derivatives is positive, and hence

$$h'(\xi) > 1 \text{ for } \xi \in (\xi_l, \xi_r)$$

This shows that h is a strictly increasing function on $[\xi_l, \xi_r]$, and therefore there exists a unique point $\xi = \xi_0 \in (\xi_l, \xi_r)$, such that $h(\xi_0) = 1$ if $h(\xi_l) < 1$ and $h(\xi_r) > 1$.

If this is the case, the system (1) has a unique fixed point. Hence we have proved the theorem. □

Theorem 3. *The location of the fixed point, provided it exists, of the system (1) is independent of the parameter θ. However, the Jacobian matrix and hence the eigenvalues ξ_i at the fixed point are dependent on θ. We have*

$$\xi_i \longrightarrow 1^- \textit{ when } \theta \longrightarrow 0^+ \textit{ for } 2 \leq i \leq n$$

while $\xi_1 = 1$ independent of θ.

Proof. The Jacobi matrix for the system (1) at a point $P = (p_1, p_2, \ldots, p_n)$ is given by

$$DF(P) = \begin{bmatrix} \frac{\partial F_1}{\partial p_1} & \frac{\partial F_1}{\partial p_2} & \cdots & \frac{\partial F_1}{\partial p_n} \\ \frac{\partial F_2}{\partial p_1} & \frac{\partial F_2}{\partial p_2} & \cdots & \frac{\partial F_2}{\partial p_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_n}{\partial p_1} & \frac{\partial F_n}{\partial p_2} & \cdots & \frac{\partial F_n}{\partial p_n} \end{bmatrix}$$

A simple calculation shows that

$$\frac{\partial F_i}{\partial p_j}(p_j) = -\frac{\theta}{n}\frac{\partial d_j}{\partial p_j}(p_j) \text{ if } i \neq j$$

and

$$\frac{\partial F_i}{\partial p_j}(p_j) = 1 + \frac{(n-1)\theta}{n}\frac{\partial d_j}{\partial p_j}(p_j) \text{ if } i = j$$

Note that

$$-\frac{\theta}{n}\frac{\partial d_j}{\partial p_j}(p_j) > 0$$

and

$$1 + \frac{(n-1)\theta}{n}\frac{\partial d_j}{\partial p_j}(p_j) < 1$$

Hence the Jacobi matrix has the form

$$DF(p) = \begin{bmatrix} 1+\frac{(n-1)\theta}{n}\frac{\partial d_1}{\partial p_1}(p_1) & -\frac{\theta}{n}\frac{\partial d_2}{\partial p_2}(p_2) & \cdots & -\frac{\theta}{n}\frac{\partial d_n}{\partial p_n}(p_n) \\ -\frac{\theta}{n}\frac{\partial d_1}{\partial p_1}(p_1) & 1+\frac{(n-1)\theta}{n}\frac{\partial d_2}{\partial p_2}(p_2) & \cdots & -\frac{\theta}{n}\frac{\partial d_n}{\partial p_n}(p_n) \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{\theta}{n}\frac{\partial d_1}{\partial p_1}(p_1) & -\frac{\theta}{n}\frac{\partial d_2}{\partial p_2}(p_2) & \cdots & 1+\frac{(n-1)\theta}{n}\frac{\partial d_n}{\partial p_n}(p_n) \end{bmatrix}$$

We will now look at the structure of this matrix, let

$$\epsilon_j = -\frac{\theta}{n}\frac{\partial d_j}{\partial p_j}(p_j)$$

then

$$\frac{\partial d_j}{\partial p_j}(p_j) = -\frac{n}{\theta}\epsilon_j$$

If $0 < \theta \ll 1$, then $0 < \epsilon_j \ll 1$, and the numbers ϵ_j are of the same magnitude for a regular family of functions $\{d_j\}$. Furthermore

$$1+\frac{(n-1)\theta}{n}\frac{\partial d_j}{\partial p_j}(p_j) = 1-(n-1)\epsilon_j$$

so

$$0 \ll 1-(n-1)\epsilon_j < 1$$

Hence the structure of the Jacobi matrix is

$$M_{J,n} = \begin{bmatrix} 1-(n-1)\epsilon_1 & \epsilon_2 & \cdots & \epsilon_n \\ \epsilon_1 & 1-(n-1)\epsilon_2 & \cdots & \epsilon_n \\ \vdots & \vdots & \ddots & \vdots \\ \epsilon_1 & \epsilon_2 & \cdots & 1-(n-1)\epsilon_n \end{bmatrix}$$

The matrix $M_{J,n}$ has (at least) one eigenvalue that equals 1. To see this, let

$$v = \left[\frac{\epsilon_n}{\epsilon_1}, \frac{\epsilon_n}{\epsilon_2}, \frac{\epsilon_n}{\epsilon_3}, \ldots, \frac{\epsilon_n}{\epsilon_{n-1}}, 1\right]^\top$$

Then $M_{J,n}v = v$, so v is an eigenvector with eigenvalue 1. To see this, let L_j denote the j-th row of $M_{J,n}$, $j = 1, \ldots, n-1$. We find that

$$\begin{aligned} L_j v &= \sum_{\substack{i=1 \\ i\neq j}}^{n} \epsilon_i \frac{\epsilon_n}{\epsilon_i} + (1-(n-1)\epsilon_j)\frac{\epsilon_n}{\epsilon_j} \\ &= (n-1)\epsilon_n + \frac{\epsilon_n}{\epsilon_j} - (n-1)\epsilon_n \\ &= \frac{\epsilon_n}{\epsilon_j} \end{aligned}$$

For $L_n v$ we find

$$L_n v = (\sum_{i=1}^{n-1} \epsilon_i \frac{\epsilon_n}{\epsilon_i}) + 1 - (n-1)\epsilon_n = (n-1)\epsilon_n + 1 - (n-1)\epsilon_n = 1$$

Now, we will resort to a perturbation argument.

We consider the matrix

$$M_{J,n} = \begin{bmatrix} 1-(n-1)\epsilon_1 & \epsilon_2 & \cdots & \epsilon_n \\ \epsilon_1 & 1-(n-1)\epsilon_2 & \cdots & \epsilon_n \\ \vdots & \vdots & \ddots & \vdots \\ \epsilon_1 & \epsilon_2 & \cdots & 1-(n-1)\epsilon_n \end{bmatrix}$$

where $1 \gg \epsilon_i > 0$. Let us assume that $\epsilon_i \approx \epsilon_j$ for all $1 \leq i, j \leq n$. Define ϵ to be the average of ϵ_i, $1 \leq i \leq n$,

$$\epsilon = \frac{1}{n} \sum_{i=1}^{n} \epsilon_i$$

and define the matrix $M_{\epsilon,n}$ by

$$M_{\epsilon,n} = \begin{bmatrix} 1-(n-1)\epsilon & \epsilon & \cdots & \epsilon \\ \epsilon & 1-(n-1)\epsilon & \cdots & \epsilon \\ \vdots & \vdots & \ddots & \vdots \\ \epsilon & \epsilon & \cdots & 1-(n-1)\epsilon \end{bmatrix}$$

Clearly $M_{\epsilon,n}$ is symmetric, that is, $M_{\epsilon,n} = M_{\epsilon,n}^\top$. From Section 3, we know that $\theta_1 = 1$ is an eigenvalue with an eigenvector $u_1 = (1, 1, 1, \ldots, 1)$. Furthermore, $\theta = 1 - n\epsilon$ is an eigenvalue of algebraic multiplicity $n - 1$. Hence, we have

$$\theta_2 = \theta_3 = \cdots = \theta_n = 1 - n\epsilon$$

We claim that the geometric multiplicity is $n - 1$. To see this, let

$$u_j = (-1, 0, 0, \ldots, 1, 0, \ldots, 0) \text{ where } 2 \leq j \leq n$$

where the first component is -1, the j-th component is 1, and all other components are 0 in u_j.

Consider $M_{\epsilon,n} u_j$, then a simple calculation shows that

$$M_{\epsilon,n} u_j = (1 - n\epsilon) u_j$$

The details are as follows, where $2 \leq j \leq n$:

$$M_{\epsilon,n} u_j = \begin{bmatrix} 1-(n-1)\epsilon & \epsilon & \cdots & \epsilon & \cdots & \epsilon \\ \epsilon & 1-(n-1)\epsilon & \cdots & \epsilon & \cdots & \epsilon \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \epsilon & \epsilon & \cdots & 1-(n-1)\epsilon & \cdots & \epsilon \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \epsilon & \epsilon & \cdots & \epsilon & \cdots & 1-(n-1)\epsilon \end{bmatrix} \begin{bmatrix} -1 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} -(1-(n-1)\epsilon)+\epsilon \\ -\epsilon+\epsilon \\ \vdots \\ -\epsilon+(1-(n-1)\epsilon \\ \vdots \\ -\epsilon+\epsilon \end{bmatrix} = \begin{bmatrix} -1+n\epsilon \\ 0 \\ \vdots \\ 1-n\epsilon \\ \vdots \\ 0 \end{bmatrix} = (1-n\epsilon) \begin{bmatrix} -1 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} = (1-n\epsilon)u_j$$

This shows that $1-n\epsilon$ is an eigenvalue with an associated eigenvector u_j for $2 \le j \le n$. Furthermore, the set $\{u_i\}_{i=1}^n$ forms a basis for $\mathbb{R}^n$, in fact $\det(U) = n$, where U is the matrix with u_i as columns. Hence we conclude that $\theta = 1-n\epsilon$ is an eigenvalue with algebraic and geometric multiplicity $n-1$.

The fixed-point equations are given by

$$\frac{1}{n}\sum_{j=1}^{n} d_j(p_j(t)) - d_i(p_i(t)) = 0$$

independent of the parameter θ.
However, we note from

$$DF(p) = \begin{bmatrix} 1+\frac{(n-1)\theta}{n}\frac{\partial d_1}{\partial p_1}(p_1) & -\frac{\theta}{n}\frac{\partial d_2}{\partial p_2}(p_2) & \cdots & -\frac{\theta}{n}\frac{\partial d_n}{\partial p_n}(p_n) \\ -\frac{\theta}{n}\frac{\partial d_1}{\partial p_1}(p_1) & 1+\frac{(n-1)\theta}{n}\frac{\partial d_2}{\partial p_2}(p_2) & \cdots & -\frac{\theta}{n}\frac{\partial d_n}{\partial p_n}(p_n) \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{\theta}{n}\frac{\partial d_1}{\partial p_1}(p_1) & -\frac{\theta}{n}\frac{\partial d_2}{\partial p_2}(p_2) & \cdots & 1+\frac{(n-1)\theta}{n}\frac{\partial d_n}{\partial p_n}(p_n) \end{bmatrix}$$

that each element in the matrix $DF(p)$ is dependent on θ. We have shown in Section 3 that $\xi_1 = 1$ is always an eigenvalue, and that the other eigenvalues are approximated by $\xi_i = 1-n\epsilon$, where $\epsilon \to 0^+$ when $\theta \to 0^+$. □

3.3. The case of strictly increasing functions

So far, we have considered strictly decreasing smooth functions $d_i : [0,1] \longrightarrow [0,1]$. However, the same types of arguments apply to strictly increasing functions.

Let $\{f_i\}_{i=1}^n$ be a family of strictly increasing smooth functions

$$f_i : [0,1] \longrightarrow [0,1] \quad \text{where} \quad i = 1,2,\cdots,n$$

Assume for simplicity that $f_i(0) = 0$ and $f_i(1) = 1$ for $1 \le i \le n$. This assumption will always ensure that all the components in the fixed-point vector are located in the interval $[0,1]$.

The dynamical system describing the update algorithm will in this case change its sign. Let $\theta > 0$, and consider the discrete system

$$p_i(k+1) = p_i(k) + \theta\left(\frac{1}{n}\sum_{j=1}^{n} f_j(p_j(k)) - f_i(p_i(k))\right) \tag{6}$$

$$\text{where} \quad i = 1,2,\cdots,n \quad \text{and} \quad \sum_{i=1}^{n} p_i = 1$$

Note here that

$$\sum_{i=1}^{n} p_i(k) = 1 \Rightarrow \sum_{i=1}^{n} p_i(k+1) = 1$$

As in the case of strictly decreasing functions, the fixed-point equations are given by

$$\frac{1}{n}\sum_{j=1}^{n} f_j(p_j) - f_i(p_i) = 0 \quad \text{where} \quad 1 \le i \le n \quad \text{and} \quad \sum_{i=1}^{n} p_i = 1$$

Hence we obtain a linear system in the quantities $f_i(p_i)$, as in the decreasing case. We find

$$f_j(p_j) = f_n(p_n) \quad \text{where } j = 1, 2, \cdots, n-1$$

The functions f_i are all invertible, and hence we can write

$$p_j = f_j^{-1} \circ f_n(p_n) \quad \text{where } j = 1, 2, \cdots, n-1$$

so we get

$$p_n + \sum_{j=1}^{n-1} p_j = 1$$

Let $\xi = p_n$. Then we have

$$\xi + \sum_{j=1}^{n-1} f_j^{-1} \circ f_n(\xi) = 1$$

We define a function h by

$$h(\xi) = \xi + \sum_{j=1}^{n-1} f_j^{-1} \circ f_n(\xi)$$

The function h is smooth and strictly increasing on $[0,1]$, so it is easy to see that $h'(\xi) > 1$. The fixed-point equations then reduce to the scalar equation

$$h(\xi) = 1$$

With the simplified conditions on the family $\{f_i\}$ we have

$$h(0) = 0 + \sum_{j=1}^{n-1} f_j^{-1}(0) = 0$$

$$h(1) = 1 + \sum_{j=1}^{n-1} f_j^{-1}(1) = n \ge 2$$

Hence there exists a unique point $\xi_0 \in [0,1]$ such that $h(\xi_0) - 1 = 0$.

We have the following theorem:

Theorem 4. *The discrete dynamical system given by* (6) *has a unique fixed point*

$$P_f = (p_1, p_2, \cdots, p_n)$$

where $0 < p_i < 1$, $1 \leq i \leq n$ *such that*

$$\sum_{i=1}^{n} p_i = 1$$

It is apparent that the matrix $DF(p)$ has the same form as in the case of decreasing functions. Hence the same arguments apply with respect to eigenvalues in this case too. Thus, we obtain the following theorem:

Theorem 5. *One of the eigenvalues of* (6) *at the fixed point is equal to 1, while the rest are less than 1 in norm, as the eigenspace corresponding to 1 is transversal to the invariant hyperplane. It follows that the fixed point is stable and attracting.*

3.4. A special case

We will consider a special case where the functions d_i, $i = 1, \ldots, n$ are of the form

$$d_i(x) = \alpha_i \exp(-\beta_i x)$$

where $0 < \alpha_i \leq 1$ and $\beta_i > 0$.

By taking the logarithm on both sides in the definition of d_i this gives

$$d_i^{-1}(x) = \frac{1}{\beta_i} \log \frac{\alpha_i}{x} \text{ where } x \in I_i$$

Furthermore, we find formally that

$$d_i^{-1}(d_n(x)) = \frac{1}{\beta_i} \left(\log \frac{\alpha_i}{\alpha_n} + \beta_n x \right)$$

Hence

$$h(\xi) = \xi + \sum_{i=1}^{n-1} \frac{1}{\beta_i} \left(\log \frac{\alpha_i}{\alpha_n} + \beta_n \xi \right) = \left(1 + \sum_{i=1}^{n-1} \frac{\beta_n}{\beta_i} \right) \xi + \sum_{i=1}^{n-1} \frac{1}{\beta_i} \log \frac{\alpha_i}{\alpha_n}$$

is a linear function in ξ.

4. Experimentation and analysis of results

In this section, we present our experimentation and analyze the results. In Section 4.1, we examine the rate of convergence of our scheme for two families of functions. While in Section 4.2, we compare our scheme to the state-of-the-art competitive scheme attributed to Stanojevic et al. (Stanojevic and Shorten, 2008, 2009a). Finally, in Section 4.3, we report some experimental results that demonstrate the applicability of our approach to the problem of rate limiting in cloud computing.

4.1. Rate of convergence

In this section, we shall provide the rate of convergence for two families of functions.

Family 1. We provide the convergence rate for a family of $d_i(x)$ functions given by:

$$d_i(x) = \exp(-\beta_i x) \tag{7}$$

For the case of this family of functions d_i, in Figure 1, we report the rate of convergence of our scheme as a function of the different values of θ, i.e, the number of iterations needed for the euclidean distance between the iterates vectors to reach the machine accuracy which is in the order of 10^{-16}.

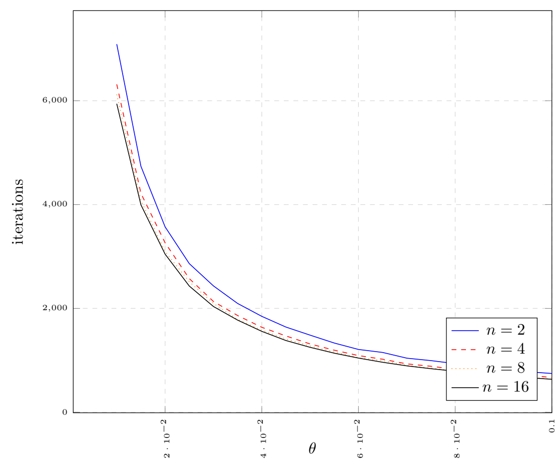


Figure 1: Rate of convergence as a function of θ for the first family of functions d_i

Family 2. We provide the convergence rate for a family of $d_i(x)$ functions given by:

$$d_i(x) = \exp(-\beta_i x^2) \tag{8}$$

For the case of this family of functions d_i, in Figure 2, we report the rate of convergence of our scheme as a function of the different values of θ, i.e, the number of iterations needed for the euclidean distance between the iterates vectors to reach the machine accuracy which is in the order of 10^{-16}.

4.2. Comparison results

To the best of our knowledge, the only available approach in the literature that treats the deterministic NEFK problem is that of Stanojevic et al. (Stanojevic and Shorten, 2008, 2009a). However, the latter approach requires some message exchange between different pairs of the components of the allocation vector. In this section, we provide some typical comparison results that show the

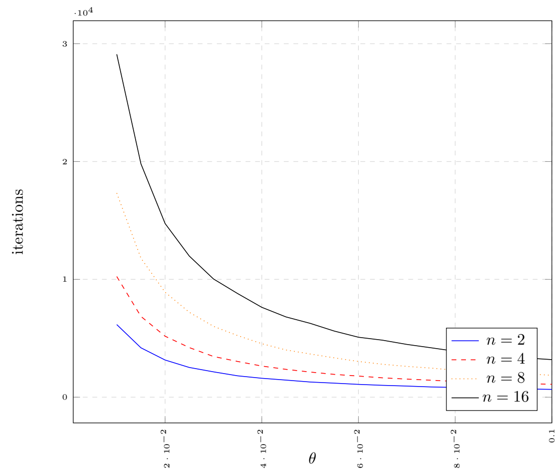


Figure 2: Rate of convergence as a function of θ for the second family of functions d_i

superiority of our approach to the work of Stanojevic (Stanojevic and Shorten, 2008, 2009a).

We choose $d_i(x)$ functions given by:

$$d_i(x) = 0.7 \exp(-ix) \tag{9}$$

In Figures 3, 4, and 5, we report the evolution of the error for both our approach and the approach taken by Stanojevic (Stanojevic and Shorten, 2008, 2009a) for $n = 4$, $n = 10$ and $n = 20$, respectively. In order to enable a fair comparison, we use the same update parameter $\theta = 0.001$ for both schemes. From Figures 3, 4, and 5, we observe that our approach is an order of magnitude faster than (Stanojevic and Shorten, 2008, 2009a).

In fact, in Figure 3, it took around 3850 iterations for the Stanojevic approach to achieve an error of less than 0.1, while our approach took around 2320 iterations to yield the same error. Similarly, in Figure 4, it took around 7705 iterations for the Stanojevic approach to reduce the error to less than 0.1, while our approach took around 2065 iterations to achieve the same result. In Figure 5, it took around 12970 iterations for the Stanojevic approach to reach an error of less than 0.1, while our approach took around 1895 iterations. Hence, according to Figure 5, our approach is almost six times faster than the Stanojevic approach in reducing the error to less than 0.1. Interestingly, it seems that the convergence speed of our approach is much less sensitive to the number of dimensions than the Stanojevic approach. In fact, the Stanojevic approach experiences a dramatic decrease in the convergence speed whenever the number of dimensions increases.

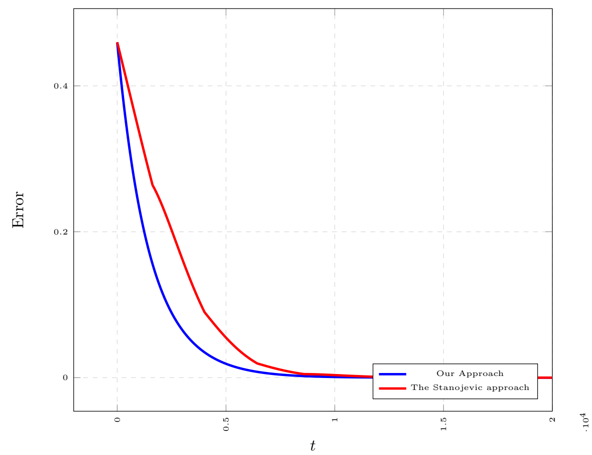


Figure 3: Evolution of the error for $n = 4$ for our approach and the Stanojevic approach

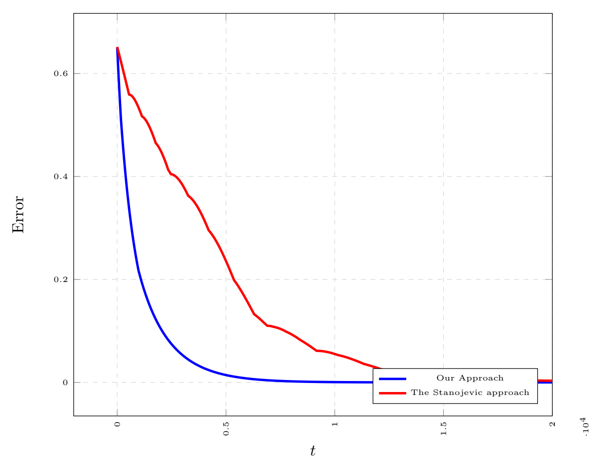


Figure 4: Evolution of the error for $n = 10$ for our approach and the Stanojevic approach

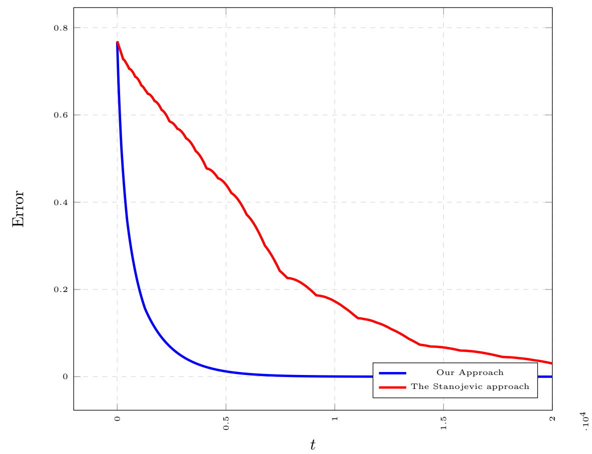


Figure 5: Evolution of the error for $n = 20$ for our approach and the Stanojevic approach

4.3. Application to rate limiting

We provide a proof of concept of our paradigm to the rate limiting problem (Doyle et al., 2012; Stanojevic and Shorten, 2008, 2009a). We resort to a set of n servers. Each limiter i is modelled as $M/M/1$ queue characterized by service rate $C_i(t)$ and packet arriving according to Poisson process with intensity λ_i.

The mean-response time at limiter i is given by:

$$MRT_i(t) = \frac{1}{C_i(t) - \lambda_i} \tag{10}$$

We follow the same line as in Stanojevic (Stanojevic and Shorten, 2009a) and instead define the "spare-bandwidth" based on estimates of the arrival rate. The spare-bandwidth will be used as the performance indicator here:

$$q_i = \lambda_i - C_i \tag{11}$$

In order to estimate λ_i, we merely use the exponential moving average approach, with learning parameter α:

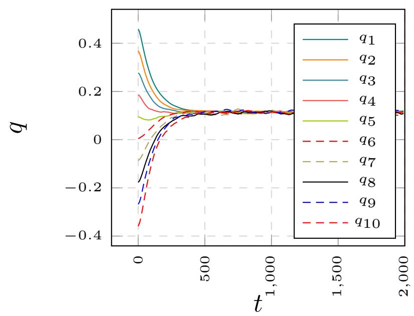
$$\hat{\lambda}_i(t+1) = (1-\alpha)\hat{\lambda}_i(t)) + \alpha\delta(t) \tag{12}$$

where $\delta(t)$ is a random variable such that $\delta(t) = 1$ if a packet arrived at the time slot t and $\delta(t) = 0$ otherwise.

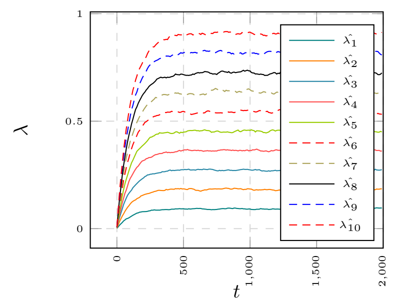
We will use the Jain's fairness index (JFI) (Bukh and Jain, 1992) to assess how equal the performance indicators $(q_1(t), ..., q_n(t))$ are. Jain's fairness is defined as:

$$JFI = \frac{(\sum_{i=1}^{n} q_i(t))^2}{n \sum_{i=1}^{n} q_i(t)^2} \tag{13}$$

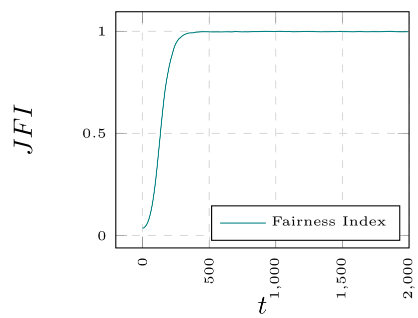
Please note that JFI lies in the unit interval. Values closer to 1 of JFI indicate fairer resource distribution.



(a) Evolution of the spare-bandwidth for $n = 10$



(b) Evolution of the arrival rate estimates for $n = 10$



(c) Evolution of the fairness index for $n = 10$

Figure 6: Dynamics of the rate limiting in a static environment for $n = 10$

In the simulation, we use $n = 10$ limiters. The demand intensity at node i is defined by:

$$\lambda_i(t) = \frac{i}{n+1} \quad \text{for} \quad i = 1, 2, \ldots, n \tag{14}$$

The aggregate service rate $C = \sum_{i=1}^{n} C_i$ is 10% larger than the aggregate traffic intensity $\sum_{i=1}^{n} \lambda_i$. We set $\alpha = 10^{-3}$ and the update parameter $\lambda = 0.1$.

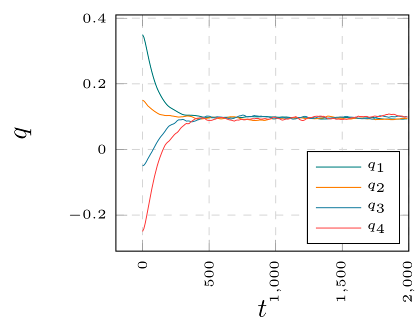
For the sake of clarity, the equations for updating the C_i are given by:

$$C_i(t+1) = C_i(t) - \lambda \left(\frac{1}{n} \sum_{i=1}^{n} q_i(C_i(t)) - q_i(C_i(t)) \right) \tag{15}$$

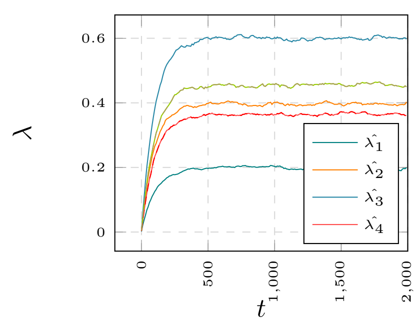
Note that C_i plays the role of p_i, and q_i plays the role of d_i.

Figures 6a and 6c show the evolution of the vector of performance indicators: $q(t) = (q_1(t), ..., q_n)$ and its JFI. Figure 6b illustrates the estimate of the traffic using our exponential moving average estimator.

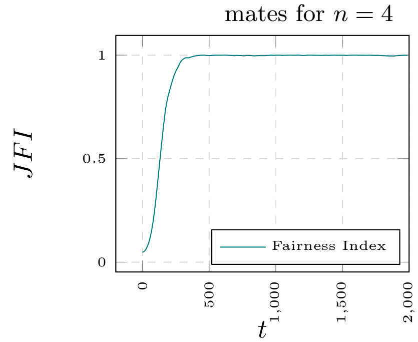
Case of $n = 4$. Similar results are given for the case of $n = 4$. We report in Figure 7a, Figure 7b and Figure 7c the evolution of spare-bandwidth, the arrival estimates, and the fairness index, respectively.



(a) Evolution of the spare-bandwidth for $n = 4$



(b) Evolution of the arrival rate estimates for $n = 4$



(c) Evolution of the fairness index for $n = 4$

Figure 7: Dynamics of the rate limiting in a static environment for $n = 10$

4.3.1. Queues in a dynamic environment

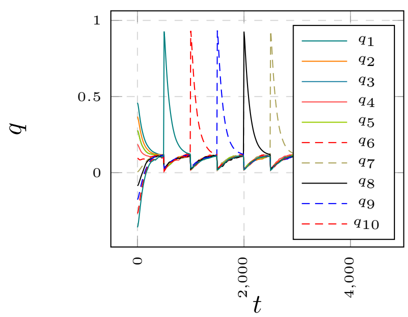
We simulate a dynamic environment where the traffic intensity is altered using a circular permutation each 500 time instants. This means that, between time instants 0 and 499, we have

$$(\lambda_1, \lambda_2, \ldots, \lambda_n) = (\frac{1}{n+1}, \frac{2}{n+1}, \ldots, \frac{n-1}{n+1}, \frac{n}{n+1})$$

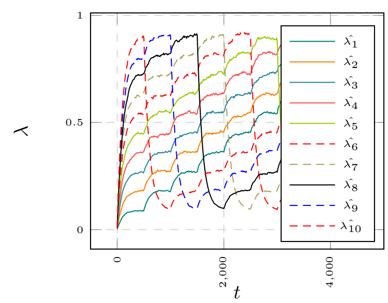
, then a circular permutation takes place at time instant 500 and the vector becomes

$$(\lambda_1, \lambda_2, \ldots, \lambda_n) = (\frac{2}{n+1}, \frac{3}{n+1}, \ldots, \frac{n}{n+1}, \frac{1}{n+1})$$

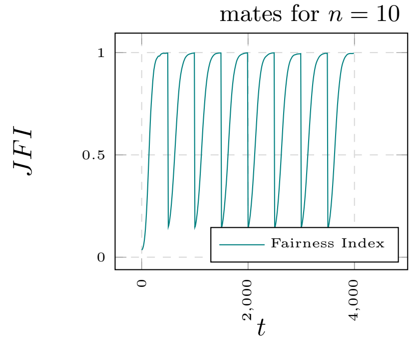
The rate limiting algorithm is run with the λ parameter specified above and the resulting $q(t)$ and $JFI(q(t))$ are shown in Figure 8c and Figure 8a. This simulation illustrates the case of changing demand patterns. We observe from the figures that the rate limiting algorithm is still able to regulate the values of $\lambda_i(t)$ yielding optimal fairness.



(a) Evolution of the spare-bandwidth for $n = 10$



(b) Evolution of the arrival rate estimates for $n = 10$



(c) Evolution of the fairness index for $n = 10$

Figure 8: Dynamics of the rate limiting in a dynamic environment for $n = 4$

4.4. Discussion

At this juncture, we draw some insightful implications based on our experimental outcomes:

- Our approach is an order of magnitude faster than the Stanojevic approach. Moreover, in contrast to the Stanojevic approach, the convergence speed of our approach is less sensitive to the increase in the number of materials (dimensions). This is illustrated in Figures 3, 4, and 5 where we increase the number of dimensions from 4 to 20 without noticing a significant decrease in the convergence speed. The informed reader would observe that the Stanojevic approach suffers from an increase in the number of exchanged messages as the number of materials increases, which slows down its convergence speed.

- When we consider the first family and the second family of functions given by equations (7) and (8) respectively, we observe that the first family achieves significantly faster convergence rate as seen from Figure 1 and Figure 2. This can be explained by the fact that the functions in the first family decrease at a much faster rate than those of the second family. This can be easily seen by considering the derivatives of (7) and (8). Loosely speaking, the functions described by the system of equations (8) are more "flat" around zero than (7). Therefore a small change in $p_i(t)$ for the first family leads to a bigger change in $d_i(t)$ and therefore to a bigger change in $p_i(t+1)$ compared to the second family.
- In Section 4.3, we consider a rate limiting application. We observe that choosing a low value of λ ($\lambda = 0.1$) yields fast convergence speed. Furthermore, the system is able to adapt to changes in a dynamic environment.

5. Conclusion

In this paper, we presented an *optimal*, efficient and yet simple solution to a class of the deterministic non-linear fractional equality knapsack (NEFK) problem that is superior to the state-of-the-art. While the majority of the state-of-the-art studies consider convex function optimization, the NEFK problem discussed here involves monotonicity of the objective function, which is a property that arises in many resource allocation problems. We provide a rigorous theoretical analysis that demonstrates the optimality of our scheme. Our simulation results illustrate the behaviour of the scheme and its optimality. An application to the problem of rate limiting in cloud computing (Stanojevic and Shorten, 2008, 2009a) is provided. Experimental results show the superiority of our scheme to the state-of-the-art (Stanojevic and Shorten, 2008, 2009a).

We shall delineate some future research directions worth pursuing:

- Fairness applications: Our scheme bears similarity to the Homo Egualis scheme (De Jong et al., 2008a) as both schemes seek achieving some type of fairness. Therefore, it is interesting to explore applying our scheme to different real-life applications where the Homo Egualis theory was applied by other researchers as well comparing both our approach and the Homo Egualis approach. Those problems include for instance fairness in multi-agent social learning (De Jong et al., 2008b) and spectrum allocation in cognitive networks (Ko et al., 2013; Xing and Chandramouli, 2008) to mention a few.

- Evolutionary game theory: Our approach can be extended to model population dynamics under decreasing fitness functions in the real of evolutionary game theory (Chen and Wang, 2009; Barari et al., 2012). More particularly, in these settings, p_i can be used to model the size of the i^{th} population while $d_i(p_i)$ can be seen as its corresponding fitness function.

- Stochastic NEFK problem: The current solution can be extended to solve the stochastic NEFK problem introduced (Granmo and Oommen, 2011, 2010b,a). The latter problem was applied for determining the optimal polling frequencies of a web crawler as well as optimal sampling for estimation with constrained resources. According to the nomenclature of the stochastic NEFK problem, p_i is a polling probability, and $d_i(p_i)$ is a binomial stochastic variable. Yazidi et al. introduced the concept of two time scale learning automata in (Yazidi et al., 2017) which also can be applied to solve the stochastic NEFK problem by estimating $d_i(p_i)$ using a faster time scale than p_i.

- Special case solution: In this paper, we treated the case where the fixed point exists and we also gave conditions in Theorem 2 that ensure its existence. However, treating the case when the optimal solution is not a fixed point of our update equations remains an open research question.

- Hierarchical solution: Under a large number of materials, we can envisage a hierarchical solution where items are divided into different groups at each level as performed in (Granmo and Oommen, 2010b, 2011; Brodowski and Podolak, 2011). The intuition is to divide the main problem into smaller sub-problems that are easier to solve. However, this needs further investigation as it is not clear how to divide the capacity between the different groups.

ACKNOWLEDGMENT

This work was supported by grants (No. TIN2013-40658-P and TIN2016-75850-R) from the FEDER funds.

References

Barari, S., Agarwal, G., Zhang, W. C., Mahanty, B., Tiwari, M., 2012. A decision framework for the analysis of green supply chain contracts: An evolutionary game approach. Expert systems with applications 39 (3), 2965–2976.

Bennani, M. N., Menasce, D., et al., 2005. Resource allocation for autonomic data centers using analytic performance models. In: Second International Conference on Autonomic Computing (ICAC 2005). IEEE, pp. 229–240.

Black, P. E., 2004. Fractional knapsack problem. Dictionary of algorithms and data structures.

Boutilier, C., Das, R., Kephart, J. O., Tesauro, G., Walsh, W. E., 2002. Cooperative negotiation in autonomic systems using incremental utility elicitation. In: Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann Publishers Inc., pp. 89–97.

Bretthauer, K. M., Shetty, B., 2002. The nonlinear knapsack problem: algorithms and applications. European Journal of Operational Research 138 (3), 459–472.

Brodowski, S., Podolak, I. T., 2011. Hierarchical estimator. Expert Systems with Applications 38 (10), 12237–12248.

Bukh, P. N. D., Jain, R., 1992. The art of computer systems performance analysis, techniques for experimental design, measurement, simulation and modeling.

Chen, Y. J., Wang, W.-L., 2009. Orders dispatching game for a multi-facility manufacturing system. Expert Systems with Applications 36 (2), 1885–1892.

de Jong, S., Tuyls, K., 2011. Human-inspired computational fairness. Autonomous Agents and Multi-Agent Systems 22 (1), 103–126.

De Jong, S., Tuyls, K., Verbeeck, K., 2008a. Artificial agents learning human fairness. In: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2. International Foundation for Autonomous Agents and Multiagent Systems, pp. 863–870.

De Jong, S., Tuyls, K., Verbeeck, K., 2008b. Fairness in multi-agent systems. The Knowledge Engineering Review 23 (2), 153–180.

Doyle, J., Shorten, R., O'Mahony, D., 2012. " fair-share" for fair bandwidth allocation in cloud computing. IEEE communications letters 16 (4), 550–553.

Dutreilh, X., Kirgizov, S., Melekhova, O., Malenfant, J., Rivierre, N., Truck, I., 2011. Using reinforcement learning for autonomic resource allocation in clouds: Towards a fully automated workflow. In: The Seventh International Conference on Autonomic and Autonomous Systems ICAS 2011. pp. 67–74.

Fulp, E. W., Ott, M., Reininger, D., Reeves, D. S., 1998. Paying for qos: an optimal distributed algorithm for pricing network resources. In: 1998 Sixth International Workshop on Quality of Service (IWQoS 98). IEEE, pp. 75–84.

Gao, L., Pan, Q.-K., 2016. A shuffled multi-swarm micro-migrating birds optimizer for a multi-resource-constrained flexible job shop scheduling problem. Information Sciences 372, 655–676.

Granmo, O.-C., Oommen, B. J., 2010a. Optimal sampling for estimation with constrained resources using a learning automaton-based solution for the nonlinear fractional knapsack problem. Applied Intelligence 33 (1), 3–20.

Granmo, O.-C., Oommen, B. J., 2010b. Solving stochastic nonlinear resource allocation problems using a hierarchy of twofold resource allocation automata. IEEE Transactions on Computers 59 (4), 545–560.

Granmo, O.-C., Oommen, B. J., 2011. Learning automata-based solutions to the optimal web polling problem modelled as a nonlinear fractional knapsack problem. Engineering Applications of Artificial Intelligence 24 (7), 1238–1251.

Granmo, O.-C., Oommen, B. J., Myrer, S. A., Olsen, M. G., 2007. Learning automata-based solutions to the nonlinear fractional knapsack problem with applications to optimal resource allocation. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 37 (1), 166–175.

He, Y.-C., Wang, X.-Z., He, Y.-L., Zhao, S.-L., Li, W.-B., 2016. Exact and approximate algorithms for discounted {0-1} knapsack problem. Information Sciences 369, 634–647.

Kellerer, H., Pferschy, U., Pisinger, D., 2004. Knapsack problems. Springer Science & Business Media.

Kephart, J. O., Chess, D. M., 2003. The vision of autonomic computing. Computer 36 (1), 41–50.

Ko, A. H., Sabourin, R., Gagnon, F., 2013. Performance of distributed multiagent multi-state reinforcement spectrum management using different exploration schemes. Expert Systems with Applications 40 (10), 4115–4126.

Loureiro, E., Nixon, P., Dobson, S., 2012. Decentralized and optimal control of shared resource pools. ACM Transactions on Autonomous and Adaptive Systems (TAAS) 7 (1), 14.

Mosk-Aoyama, D., Roughgarden, T., Shah, D., 2010. Fully distributed algorithms for convex optimization problems. SIAM Journal on Optimization 20 (6), 3260–3279.

Palomar, D. P., Chiang, M., 2006. A tutorial on decomposition methods for network utility maximization. IEEE Journal on Selected Areas in Communications 24 (8), 1439–1451.

Stanojevic, R., Shorten, R., 2008. Fully decentralized emulation of best-effort and processor sharing queues. In: ACM SIGMETRICS Performance Evaluation Review. Vol. 36. ACM, pp. 383–394.

Stanojevic, R., Shorten, R., 2009a. Generalized distributed rate limiting. In: International Workshop on Quality of Service (IWQoS).

Stanojevic, R., Shorten, R., 2009b. Load balancing vs. distributed rate limiting: an unifying framework for cloud control. In: IEEE International Conference on Communications. IEEE, pp. 1–6.

Stanojevic, R., Shorten, R., 2010. Distributed dynamic speed scaling. In: 29th IEEE International Conference on Computer Communications (INFOCOM).

Subramoniam, K., Maheswaran, M., Toulouse, M., 2002. Towards a micro-economic model for resource allocation in grid computing systems. In: IEEE Canadian Conference on Electrical and Computer Engineering (CCECE). Vol. 2. IEEE, pp. 782–785.

Wang, X., Du, Z., Chen, Y., Li, S., 2008. Virtualization-based autonomic resource management for multi-tier web applications in shared data center. Journal of Systems and Software 81 (9), 1591–1608.

Xing, Y., Chandramouli, R., 2008. Human behavior inspired cognitive radio network design. IEEE Communications Magazine 46 (12).

Yazidi, A., Hammer, H. L., Jonassen, T. M., 2017. Two-timescale learning automata for solving stochastic nonlinear resource allocation problems. In: International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. Springer, pp. 92–101.