International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.

http://ijpsat.ijsht-journals.org
Vol. 29 No. 1 October 2021, pp.558-564

Nash Bargaining on a Processor Sharing Queue

Toky Basilide RAVALIMINOARIMALALASON[1], Mirisoa RAKOTOMALALA[2], Falimanana RANDIMBINDRAINIBE[3]

[1,2,3]Ecole doctorale en Sciences et Techniques de l'Ingénierie et de l'Innovation
Ecole Supérieure Polytechnique d'Antananarivo - Université d'Antananarivo - Madagascar

***Abstract* – This paper presents an approach through the bargaining game theory on the capacity sharing in a queue with processor sharing discipline. It has a brief review of such a type of queue and the Nash bargaining solution. It demonstrates the feasibility of insensitive and balanced sharing. It also proves that sharing is not insensitive and balanced if the notion of priority is integrated in the queue.**

***Keywords* – Game theory, Nash, Bargaining, Queue, Processor sharing**

I. Introduction

The processor sharing is a limit of the Round-Robin scheduling when the time step tends towards 0. It knows a great success because it shares all available resources across customers. The term queue has become a misnomer because in reality there is no more queue training, all arriving customers are immediately served with part of the system's resources [1]. Many authors have studied this kind of queue discipline, especially its insensitive characteristic to the customer's arrival distribution or their service distribution [1][2][3][4].

What interests us in this paper is rather how to share the available resources to customers. We already focused on resource sharing even they are discrete [5] or fractional [6] but here we based the sharing on game theory, which is a very interesting field to study or even to predict the outcome of a conflict situation such as the sharing of resources in a processor sharing queue. We assimilate the resource sharing as a bargaining issue between customers in the queue.

II. Processor sharing queue

A. Description

The "Processor Sharing" (PS) service discipline is a discipline that simultaneously shares capacity of a server among all clients requesting services from that server.

A queue that adopts this discipline to schedule its clients is called a processor sharing queue or simply a PS queue. Thus, for fair sharing, at time *t* a customer is served at speed $C/x(t)$ if $x(t)$ customers are present in a PS queue with capacity *C*.

A queueing network is said to processor sharing if all of the queues in this network are PS queues. A network of PS queues can therefore be interpreted as a network of queues where each i-th PS queue has a capacity $\varphi_i(\mathbf{x})$ which may depend on the number $\mathbf{x}$ of customers in the network.

The network state **x** is a vector which components are the number of customers at each node of the network : $\mathbf{x} = (x_1, x_2, ..., x_n)$ where x_i is the number of customers at the node *i*.

We call $\varphi_i(\mathbf{x})$ the capacity allocation at state x. It is assumed that the services cannot be interrupted, ie it is supposed that if $x_i > 0$ then $\varphi_i(\mathbf{x}) > 0$ for all *i*.

Arrivals on node *i* is a Poisson process with intensity λ_i. Customers service durations of the queue *i* at node *i* are independent and identically distributed following exponential distribution with average $1/\mu_i$.

B. Balance and insensitivity

Let's consider a queueing network of *n* queues $(n \in \mathbb{N}^*)$ where each queue *i* has a capacity $\varphi_i(\mathbf{x})$ depending to the network state **x**. Capacities of such network are called balanced if :

$$\varphi_i(\mathbf{x})\varphi_j(\mathbf{x}-\mathbf{e}_i) = \varphi_j(\mathbf{x})\varphi_i(\mathbf{x}-\mathbf{e}_j), \qquad i, j = 1, ..., n \tag{1}$$

where $\mathbf{e}_i$ denotes a vector whose *i*-th component is 1 and the others are all zero.

This equality in (1) states that the change in the allocation to queue *i* when a customer in queue *j* leaves its queue is identical to the change in the allocation to queue *j* when a customer in queue *i* also leaves its queue.

Considering a direct path from state **x** to **0**, $\langle x, x-e_{i_1}, ..., x-e_{i_1}-...-e_{i_{C-1}}, 0\rangle$, which has a length *c* where $c = x_1 + ... + x_n$ is the total number of customers in a state **x**. We define a balance function Φ expressed by [7] :

$$\Phi(\mathbf{x}) = \frac{1}{\varphi_{i_1}(\mathbf{x})\varphi_{i_2}(\mathbf{x}-\mathbf{e}_{i_1})...\varphi_{i_C}(\mathbf{x}-\mathbf{e}_{i_1}-\mathbf{e}_{i_{C-1}})} \tag{2}$$

The balance property requires that the balance function is independent of the path from state **x** to state **0**. The capacities $\varphi_i(\mathbf{x})$, for all *i*, are characterized by the balance function Φ :

$$\varphi_i(\mathbf{x}) = \frac{\Phi(\mathbf{x}-\mathbf{e}_i)}{\Phi(\mathbf{x})} \tag{3}$$

We say that the capacities are balanced by the function Φ.

The stationary distribution $\pi(\mathbf{x})$ of the number of customers in the network can be expressed from this balance function by the relation:

$$\pi(\mathbf{x}) = K\Phi(\mathbf{x})\prod_{i=1}^{I}\left(\frac{\lambda_i}{\mu_i}\right)^{x_i} \tag{4}$$

where *K* is a normalization constant according to the probabilistic definition of function π.

We see that this function does not depend on the distribution of customers' arrival nor their service requests but only by their averages. In this case, we say that the network is said to be insensitive. [1][3][8][9]

III. Bargaining game

A. Description of a bargaining game

People are now faced with such a set of alternatives that their preferences are in conflict. Among them, one can serve as a reference that we will call the status quo or point of disagreement, but of course, there are other preferences that can give more than the status quo. There is potentially an interest in not just starting a negotiation which should lead to a consensus on a particular alternative. It is an alternative that gives more to that person but to the detriment of others. Two cases can arise: Either the negotiation is successful, or the opposite and in this case, the status quo is maintained. This point plays a role of threat for these people. This kind of game is called a bargaining game, which is one branch of game theory.

Nash described a solution that predicts the outcome of such bargaining.

B. A bargaining problem

A bargaining problem is defined by the couple (U,d) where $d \in U$ and U a subset of $\mathbb{R}^n$ satisfying the following properties:

- *U* is convex and closed
- *U* is bounded: $\exists \overline{u} \in \mathbb{R}^n$ such as $u_i < \overline{u}_i$ for all $i = 1$ to n and for all u of U.
- *d* is dominated in the sense of Pareto : $\exists u \in U$ such as $u_i > d_i$ for all $i = 1$ to n

U is nonempty because it contains at least the point of disagreement or the status quo. The convexity of *U* reflects the fact that the set of alternatives is a continuum and that the players do not have a disposition for risk. *U* bounded guarantees a limit on players' gain, and Pareto dominance invokes that potential gain really exists.

A solution to this bargaining problem is a rule described by a function *f* such that any bargaining problem (*U*, *d*) associates the element $\overline{u} = f(U,d)$ of the set *U*. $\overline{u}_i = f_i(U,d)$ where f_i defines the utility level for each *i*.

The solutions to this bargaining problem must meet certain conditions. [10]

C. Nash solution

Nash's solution is based on axioms which lead to the uniqueness of the solution. This corresponds to the maximization of the function $f(u)$ on the set U :

$$f(u) = \prod_{i=1}^{n} \left(u_i - d_i\right)^{\omega_i} \tag{5}$$

where ω_i denotes an eventual weight for a player *i* of the bargaining.

This function is continuous over a compact set, so the solution exists. The uniqueness is that the set *U* is convex and the function *f* is strictly concave, which results in contour curves that are strictly convex with respect to the origin. [11]

IV. Resource bargaining in a PS queue

A. Game formulation

Let's study the case of a single server of PS queue. Customers in this queue are divided into customer class numbered from 1 to *n*. This queue can be assimilated to *n* parallel queues, each queue *i* of which serves a class *i* of customers. At any time, each queue *i* serves x_i customers and the system state is the vector $\mathbf{x} = \left(x_1, x_2, ..., x_n\right)$.

The entire system then hosts $|\mathbf{x}|$ customers where $|\mathbf{x}|$ denotes the sum of all the components of the vector **x**.

The sum of the individual capacities $\varphi_i\left(\mathbf{x}\right)$ of each queue *i* depending on the state **x** of the system is therefore equal to the total capacity *C* available for the system :

$$\sum_{i=1}^{n} \varphi_i\left(\mathbf{x}\right) = C \tag{6}$$

For each queue i, the available resource $\varphi_i\left(\mathbf{x}\right)$ is shared equally between the x_i customers. Then, the available resource for a class *i* customer is:

$$r_i = \frac{\varphi_i(\mathbf{x})}{x_i} \tag{7}$$

We are going to model this PS queues network to a game in which the players are the customers of the network who bargain the available resources on the entire network. Each queue *i* has the utility function $f(\varphi_i(\mathbf{x}))$.

B. Solution of the bargaining problem

The solution of this bargaining problem is :

$$\begin{aligned} &\arg\max_{\varphi_i} \sum_{i=1}^{n} x_i . f(\varphi_i) \\ &\text{under constraint : } \sum_{i=1}^{n} \varphi_i = C \end{aligned} \tag{8}$$

The term $x_i . f(\varphi_i)$ denotes that the utility on queue *i* is common to all x_i customers, and the overall utility is calculated across all of those customers.

This leads to a convex optimization problem, and the resolution uses Lagrange multipliers with Karush – Kuhn – Tucker (KKT) conditions. The associated function to find the Lagrange multipliers is given by:

$$L(x, \lambda) = \sum_{i=1}^{n} x_i . f(\varphi_i) + \lambda . \left(\sum_{i=1}^{n} \varphi_i - C \right) \tag{9}$$

The utility function f can be chosen so that the resulting solution is equal to Nash's solution with the origin as the point of disagreement. In this case, it suffices to use a logarithm function:

$$f(\varphi_i) = \omega_i . \log(\varphi_i) \tag{10}$$

The choice of this function derives from the efficiency function which gives the proportion of capacity obtained by queue *i* compared to the total capacity of the system *C*: $\log(\varphi_i) / \log(C)$.The term ω_i denotes a weight of each customer class *i*.

The associated Lagrange function is therefore:

$$L(x, \lambda) = \sum_{i=1}^{n} x_i . \omega_i . \log(\varphi_i) + \lambda . \left(\sum_{i=1}^{n} \varphi_i - C \right) \tag{11}$$

Among the KKT conditions, for all *k*, we have:

$$\frac{x_k . \omega_k}{\varphi_k} + \lambda = 0 \tag{12}$$

Then:

$$\begin{aligned} &\frac{x_i . \omega_i}{\varphi_i} = \frac{x_j . \omega_j}{\varphi_j} \qquad \forall i \neq j \\ &\text{and } \sum_{i=1}^{n} \varphi_i = C \end{aligned} \tag{13}$$

Solving this system of equations shows that the capacity that must be allocated to queue *i* is:

$$\varphi_i(\mathbf{x}) = \frac{x_i \omega_i}{\sum_{k=1}^{n} x_k \omega_k} . C \tag{14}$$

At first glance, this is a kind of allocation proportional to the number of customers in each queue and the weights assigned to each queue.

C. Balance and insensitivity

According to the capacity allocated to each queue in (14), we have:

$$\varphi_i(\mathbf{x}) = \frac{x_i \omega_i}{\sum_{k=1}^{n} x_k \omega_k}.C \tag{15}$$

$$\varphi_j(\mathbf{x}) = \frac{x_j \omega_j}{\sum_{k=1}^{n} x_k \omega_k}.C \tag{16}$$

$$\varphi_i(\mathbf{x}-\mathbf{e}_j) = \frac{x_i \omega_i}{\left(\sum_{k=1}^{n} x_k \omega_k\right) - \omega_j}.C \tag{17}$$

$$\varphi_j(\mathbf{x}-\mathbf{e}_i) = \frac{x_j \omega_j}{\left(\sum_{k=1}^{n} x_k \omega_k\right) - \omega_i}.C \tag{18}$$

The balance equation (1) gives us:

$$\frac{1}{\left(\sum_{k=1}^{n} x_k \omega_k\right) - \omega_j} = \frac{1}{\left(\sum_{k=1}^{n} x_k \omega_k\right) - \omega_i} \tag{19}$$

This equality is true if and only if $\omega_i = \omega_j$ for all $i \neq j$.

That is, we must assign the same weight to all queues to have a balanced and insensitive system. In other words, this shows that the fact of endowing the system with the notion of priority on the basis of weights removes its characteristic of balance and insensitivity.

Of course, more resources must always be allocated to the higher priority class queue and the notion of balance defined in the second paragraph will no longer be maintained.

D. Stationary distribution

In the following, we will study a balanced and insensitive system, with a capacity of each queue equal to:

$$\varphi_i(\mathbf{x}) = \frac{x_i}{\sum_{k=1}^{n} x_k}.C = \frac{x_i}{|\mathbf{x}|}.C \tag{20}$$

We have:

$$\varphi_i(\mathbf{x}) = \frac{\Phi(\mathbf{x}-\mathbf{e}_i)}{\Phi(\mathbf{x})} = \frac{x_i}{|\mathbf{x}|}.C \tag{21}$$

Therefore:

$$\Phi(\mathbf{x}) = \Phi(\mathbf{x}-\mathbf{e}_i)\frac{|\mathbf{x}|}{C.x_i} \tag{22}$$

$$\Phi\left(\mathbf{x}-\mathbf{e}_i\right)=\Phi\left(\mathbf{x}-2\mathbf{e}_i\right)\frac{\left|\mathbf{x}\right|-1}{C.\left(x_i-1\right)} \tag{23}$$

$$\Phi\left(\mathbf{x}-x_i\mathbf{e}_i\right)=\Phi\left(\mathbf{x}-\left(x_i+1\right)\mathbf{e}_i\right)\frac{\left|\mathbf{x}\right|-x_i}{C} \tag{24}$$

Thus:

$$\Phi\left(\mathbf{x}\right)=\Phi\left(\mathbf{x}-x_i\mathbf{e}_i\right)\frac{\left|\mathbf{x}\right|...\left(\left|\mathbf{x}\right|-x_i\right)}{C^{x_i}.x_i!} \tag{25}$$

Doing the same for all i, we have:

$$\Phi\left(\mathbf{x}\right)=\Phi\left(\mathbf{0}\right)\frac{\left|\mathbf{x}\right|!}{C^{\left|\mathbf{x}\right|}.\left(x_1!...x_n!\right)} \tag{26}$$

And the stationary distribution follows:

$$\pi\left(\mathbf{x}\right)=K.\frac{\left|\mathbf{x}\right|!}{C^{\left|\mathbf{x}\right|}.\left(x_1!...x_n!\right)}.\prod_{i=1}^{n}\rho_i^{x_i} \tag{27}$$

where K is a normalization constant obtained from the probabilistic definition of π : $\sum\pi=1$, and $\rho_i=\frac{\lambda_i}{\mu_i}$ is the traffic intensity at queue i.

V. Conclusion

The bargaining game theory approach discussed in this work has allowed us to share the resources of a processor sharing queue.

This method with Nash's solution allowed us to find out how to share these resources across classes of customers. This approach also allowed us to demonstrate that the notion of priority in the PS queue makes it lose the notion of insensitivity and balance in terms of sharing this resource.

References

[1] T. Ravaliminoarimalalason, Study of the Whittle network and its application to model packet switched networks (Etude du réseau de Whittle et son application pour modéliser les réseaux à commutation de paquet). DEA Cognitive sciences and applications, ESPA – University of Antananarivo, 2013.

[2] A.T. Minh, Insensitivity in queuing networks and applications to the sharing of computer resources (Insensibilité dans les réseaux de files d'attente et applications au partage de ressources informatiques). Network and Communications Theory Research Group, Doctorate Thesis, 2007.

[3] A. Proutière, Insensitivity and stochastic bounds in queuing networks (Insensibilité et bornes stochastiques dans les réseaux de files d'attente). France Telecom R&D – Ecole Polyechnique, Doctorate Thesis, 2003.

[4] K. Abhay, J. Parekh, A generalized processor sharing approach to flow control in integrated services networks. The Johns Hopkins University and Massachusetts Institute of Technology, Doctorate Thesis, 1992.

[5] T.B. Ravaliminoarimalalason, F. Randimbindrainibe, "Distribution of Occupied Resources on A Discrete Resource Sharing in A Queueing System", International Journal of Engineering Research & Technology, ISSN 2278-0181, pp 638-643, Vol. 10 Issue 1, 2021

[6] T.B. Ravaliminoarimalalason, M. Rakotomalala, F. Randimbindrainibe, “Distribution of occupied resources on a fractional resource sharing in a queueing system”, International Journal of Operational Research, DOI 10.1504/IJOR.2021.10039107, 2021

[7] T.B. Ravaliminoarimalalason, F. Randimbindrainibe, From queues to teletrafics (Des files d'attente aux télétrafics). Edition ISTE, 244 pp, 2021.

[8] T.T.H. Dao, Zero-automatic queues and networks. Laboratory of Algorithmic Computing, University Paris Diderot, Doctorate Thesis, 2007.

[9] A.R. Ward, Predicting response times in processor sharing queues. Dpt of Management Science and Engineering - Stanford University, AT&T Labs Shannon Laboratory, 2000.

[10] P. Dehez, Conflict, bargaining, sharing, power (Conflit, marchandage, partage, pouvoir). Core-UCL, 2007.

[11] C.O. Alexander, W. Ledermann, “The constrained Nash Bargaining solution”, J. Opl Res. Soc., Vol. 45, University of Sussex, 1994.