International Journal of Analysis and Applications
Volume 16, Number 1 (2018), 125-136
URL: https://doi.org/10.28924/2291-8639
DOI: 10.28924/2291-8639-16-2018-125

MAXIMIZE MINIMUM UTILITY FUNCTION OF FRACTIONAL CLOUD COMPUTING SYSTEM BASED ON SEARCH ALGORITHM UTILIZING THE MITTAG-LEFFLER SUM

RABHA W. IBRAHIM

Department of Mathematics and Computer Science, Modern College of Business & Science, 133, Muscat, Oman

Corresponding author: rabhaibrahim@yahoo.com

Abstract. The maximum min utility function (MMUF) problem is an important representative of a large class of cloud computing systems (CCS). Having numerous applications in practice, especially in economy and industry. This paper introduces an effective solution-based search (SBS) algorithm for solving the problem MMUF. First, we suggest a new formula of the utility function in term of the capacity of the cloud. We formulate the capacity in CCS, by using a fractional diffeo-integral equation. This equation usually describes the flow of CCS. The new formula of the utility function is modified recent active utility functions. The suggested technique first creates a high-quality initial solution by eliminating the less promising components, and then develops the quality of the achieved solution by the summation search solution (SSS). This method is considered by the Mittag-Leffler sum as hash functions to determine the position of the agent. Experimental results commonly utilized in the literature demonstrate that the proposed algorithm competes approvingly with the state-of-the-art algorithms both in terms of solution quality and computational efficiency.

1. Introduction

The utility is a quantity of favorites over some set of properties; it signifies fulfillment skilled by the agent of a property. The notion is a significant under heading of the optimization theory in the economy and

Received 24th September, 2017; accepted 6th December, 2017; published 3rd January, 2018.

2010 *Mathematics Subject Classification.* 44A45, 34A08, 26A33.

Key words and phrases. fractional calculus; fractional differential equation; fractional differential operator; utility function; cloud computing system .

the game theory: since the agent cannot straight measure benefit, fulfillment or gladness from the property or service, the economists have developed methods of signifying and measuring the utility in terms of the measurable economic selections. One of the challenges in CCS is how to dynamically distribute information (data) to the agents such that qualities of service constraints are fulfilled and operating costs are minimized. The adjustment between these two contradictory aims can be expressed by a utility function. The general formula of the linear utility function in CCS is as follows (see [1])

$$U_i = \sum_{j=1}^{n} \omega_j \rho_{ij},$$

where ρ_{ij} is the rate of outcomes and ω_j is the weight achieving $\sum_j \omega_j = 1$. The non-linear utility function is defined by the formula (see [2])

$$U_i = \prod_{j=1}^{n} \rho_{ij}^{\omega_j}, \quad 0 < \omega_j < 1.$$

The author introduced a modification to the non-linear utility function by employing the Tsalis entropy [3], Wiener processing [4] and by using the solution of fractional differential equation [5]. In our discussion, we impose a new formula of the non-linear utility function involving fractional calculus. The new formula is based on the classic definition of the Riemann-Liouville operators, where the integral operator is given by (see [6]- [9])

$$I^{\alpha}(\nu)(t) = \frac{1}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} \nu(s)\, ds$$

and the fractional differential operator (the classic fractional calculus) is formulated by

$${}_0D_t^{\alpha}\nu(t) = \frac{d}{dt}\, I^{1-\alpha}\nu(t), \quad \alpha \in (0,1).$$

This formula is modified recent active utility functions. The principle of the definition is based on the diffeo-integral diffusion fractional equation. Experimental results commonly utilized in the literature demonstrate that the proposed algorithm competes approvingly with the state-of-the-art algorithms both in terms of solution quality and computational efficiency.

1.1. **Aims.**

- **Purpose**: The paper schemes self-possessed of multiple networking agents within a location. It proposes a mathematical modeling of intelligence CCS management systems. The study aims to Maximize minimum utility function of the system in term of the capacity of the cloud. The study of this problem is very limited and is not suggested widely in the CCS.
- **Methodology:** The paper suggested a multi-agent system based on the fractional calculus of a hybrid diffeo-integral system. The continuous dynamics are established by employing the fractional differential equation and the fractal is considered for a time, while the space is illustrated in ordinary cases. The technique is based on the Mittag-Leffler sum (diffusion processes). This process is the

net movement of information or data from a state of low observation in a state of high observation. This property is a basic realization in the multi-agent computing systems. Moreover, this process yields a convergence to the equilibrium point (a point of maximal utility). Also, can be suggested as a summation search solution (SSS).

- **Outcomess**: The paper provides a large number of solutions to maximize the minimum utility function in CCS. The method is a proficient technique to solve various types of fractional differential systems.
- **Research modulation/implications**: Researchers are encouraged to test and modify the proposed method, using any mathematical inclusion containing the capacity of the cloud. One can add a controller to the system, with less constrains.
- **Practical implications**: The paper involves inspirations for the expansion of a powerful for carrying and integrating the cloud system stability and movement.
- **Originality/value**: This paper satisfies an acknowledged need to study how to the utility function of cloud computing system can be maximized under the same capacity of the system. Here, we do not formulate the dynamic system of the cloud as can be recognized in the formerly works.

2. Algorithm

Our algorithm is divided into two parts. The first part deals with derivative the utility function from the fractional diffeo-integral equation. The second part contains the numerical computation, based on the summation search solution (SSS). We introduce an effective SSS algorithm for solving the problem MMUF.

2.1. **The diffeo-integral equation.** In this study, we consider the transport of information or data, in CCS in one-dimensional with the capabilities of real-time observing of the saturated line average. We derive a mathematical model based on data balance that incorporates dispersion and diffusion. Let x_i, $i = 1, ..., n$ be the need (what the agent needs from CCS to receive information) of the user (agent) i in the sector Λ_i from CCS, $x = (x_1, ..., x_n)$ be the set of demand (acts or positions) for all users; the total cross-sectional area normally available for flow is $\omega\Lambda = \sum_{i=1}^{n} \omega\Lambda_i$. Now let $C(x,t)$ be the capacity of CCS during time t with its flux $Q(x,t)$. We additionally assume that the sensor is produced or shattered with rate $F(x,t)$. For example, F could be a reaction term or source term. Finally, we denote by V the Darcy velocity and $\upsilon = V/\omega$, $\omega \in (0,1)$ the average velocity. All the above functions are in $C[\mathbb{R}^n \times J, \mathbb{R}]$ (the space of real valued functions). To introduce the fractional diffeo-integral equation that imposes the utility function, we let $x_i \in [a_i, b_i]$ and $\Omega := \prod_{i=1}^{n} [a_i, b_i]$ then we have the following equation:

$$ {}_0D_t^{\alpha}\Big(\int_{\Omega} C(x,t)\omega\Lambda dx\Big) = Q(\underline{x},t)\Lambda - Q(\bar{x},t)\Lambda + \int_{\Omega} F(x,t)\Lambda dx, \tag{2.1}$$

where $\underline{x}$ is the minimum value of $x \in \Omega$ and $\bar{x}$ is the maximum value of $x \in \Omega$.

The term on the left-side is the rate of change of the total amount of sensor in the sector, and the first two terms on the right compute the rate that the sensor flows into the sector at $\underline{x}$ and the rate that it flows out at $\bar{x}$; the last term is the rate that the sensor is established in the sector. By the continuity of the functions, we have

$$\int_{\Omega} \left(C_t^{(\alpha)} \omega + Q_x - F \right) dx = 0,$$

where $C_t^{(\alpha)}(x,t) := {}_0D_t^{\alpha} C(x,t)$. Consequently, we obtain

$$C_t^{(\alpha)} \omega + Q_x = F.$$

There are three generally accepted methods to determine the function $Q(x,t)$. It can be calculated by advection (the transfer of the flow of a fluid, especially horizontally in the atmosphere, the sea or the cloud), which incomes that users are simply approved by the CCS; this brings us to describe Q_a in the following formal:

$$Q_a(x,t) = VC(x,t).$$

Another technique of utilization is by diffusion. This is the distribution initiated by the random signal and crashes by the users themselves. In this case the function Q_d is represented by

$$Q_d(x,t) = -\omega \delta C_x(x,t),$$

where δ is the diffusion coefficient. There is a third type of the function Q, which is called dispersion. This case the flux will consider to be hybrid Q_h, taking the formula

$$Q_h(x,t) = -\omega \varphi C_x(x,t),$$

where φ is the dispersion coefficient. Therefore, we have the general form of the flux as follows:

$$Q(x,t) = Q_a(x,t) + Q_d(x,t) + Q_h(x,t).$$

This implies the fractional reaction-advection-dispersion equation

$$\omega C_t^{(\alpha)}(x,t) = \left(\omega \Delta C_x \right)_x (x,t) - VC_x(x,t) + F(x,t), \quad \Delta = \varphi + \delta,$$

which can be written as a fractional differential equation in the form:

$$C_t^{(\alpha)}(x,t) = \Delta C_{xx}(x,t) - \upsilon C_x(x,t) + \frac{1}{\omega} F(x,t), \quad \omega \in (0,1).$$

Hence, the general solution of the above equation is formulated by

$$C(x,t) = U(x, \epsilon t^{\alpha}), \quad \alpha \in (0,1], \tag{2.2}$$

where U is represented in the fractional utility function. Given a set $N = \{1,2,...,n\}$ of $n-$ agents and the utility of CCS is denoted by $u_{ij} \geq 0\,(I,j)$. A dispersion problem consists in choosing a set M of agents from

N, such that some objective function defined on the selected agents of M is maximized or minimized the utility of CCS. Formally, the Max-Minsum utility function (MMUF) can be stated as follows:

$$\begin{aligned} &Maximize\ Min U(x, \epsilon t^{\alpha}) = \sum_{j \in M} u_{ij} \\ &\text{subject to, } \Theta := \{M \subset N,\ |M| = m\}. \end{aligned} \tag{2.3}$$

$$\Big(\alpha \in (0,1], \quad \epsilon \in (0,1), \quad t \in J = [0,T], T < \infty\Big).$$

Our aim is to find a suitable solution in terms x and t^{α}. In other words, we look for formal power series. Since our approach depends on the fractional calculus, therefore, we suggest applying the Mittag-Leffler sum.

A system to formalize the notion of formal preferences is by transmission to each state x a real number $U(s)$ that is called the utility of state x for that particular agent. Officially, for two states x and x' holds $U(x) > U(x')$ if and only if the agent selects state x to state x', and $U(x) = U(x')$ if and only if the agent is unmoved between x and x'. Next subsection describes the numerical approach of it.

2.2. **The Mittag-Leffler sum.** The Mittag-Leffler function occurs obviously in the solution of fractional integral equations and mainly in the investigation of the fractional generalization of the kinetic equation, random walks, Lvy flights, and so-called super-diffusive transport. The ordinary and generalized Mittag-Leffler functions include between a purely exponential law and power-like performance of phenomena administered by ordinary kinetic equations and their fractional complements [10].

Let

$$u(x,t) = \sum_{n=0}^{\infty} u_n \left(t^{\alpha} x\right)^n$$

be a formal power series in t and x. Define the transform $\mathcal{B}_{\alpha} u$ of u by

$$\mathcal{B}_{\alpha} u(x,t) \equiv \sum_{n=0}^{\infty} \frac{(t^{\alpha} x)^n}{\Gamma(1+\alpha n)}. \tag{2.4}$$

Then the Mittag-Leffler sum of u is given by $\lim\limits_{\alpha \to 0} \mathcal{B}_{\alpha} u(x)$ if each sum converges and the limit exists. Then the Mittag-Leffler sum (MLS) of u is given by

$$\mathrm{ES}_{\alpha} u(x) = \int_0^{\infty} e^{-t} \mathcal{B}_{\alpha} u(t^{\alpha} x)\, dt.$$

In other words, the utility function is maximized by using the transform $\mathcal{B}_{\alpha}$. Since the utility function is formalized by the capacity of the CCS, then the capacity of the cloud will not reduce.

2.3. **The objective function.** The basic support learning model contains the following facts: a set includes environmental data and agents' positions S; a set contains the movements M; strategies of transitioning from the set of agents' positions to the set of movements; procedures that control the scalar immediate objective (SIO) of a move (shift); and procedures that define what the agent supports.

The procedures are often stochastic. The observation naturally includes the SIO related to the last change. In this work, the agent is also expected to detect the present environmental formal, in which situation we talk about full observability, whereas in the contradictory situation, we focus about partial observability (see Fig.1). Occasionally, the set of movements (M) existing to the agent is controlled.

A support learning agent relates to its environment in discrete time steps. At each time t, the agent collects an observation o_t, which naturally includes the SIO (x_t). It then selects a movement m_t from the set of movements (M), which is consequently sent to the environment. The environment transfers to a new formal x_{t+1} and the object x_{t+1} related with the transition (x_t, m_t, x_{t+1}) is calculated. The aim of a supportive learning agent is to gather as much reward as potential. The agent can select any movement as a function of the past and it can even randomize its achievement selection.

When the agent's act is associated with an agent which acts optimally for the beginning, the variation in act implies rise to the notion of regret. Annotation is that in order to act near optimally, the agent must aim about the long period significance of its actions (i.e., maximize the forthcoming utility). In this position, we can define the formal of MMUF by the criteria

$$U^*(x,t) = \max_{t \in J} \mathcal{B}_\alpha U(x,t), \tag{2.5}$$

where $\mathcal{B}_\alpha$ is formulated in (2.4).

3. Simulation

Jafari et al [11] and Ashouraie and Navimipour [12] utilized two types of algorithm to determine the utility function in CCS. They applied the genetic algorithms and heterogeneous resources, respectively. Salih et al., [1, 2] introduced a method of employing the user preferences. Fractional utility functions, for the first time are suggested by the authors in [3]- [5]. We deal with computational agents; communication involves several stages of abstraction. On the lower, 'network' stage, one would make sure that the messages that are communicated among the agents arrive safely and timely at their destination. A formal technique to designate communicate is by giving each communication as an action that updates the information of an agent about aspects of the state; like those described in Fig.1. The communication primitives that are switched among agents are normally raised to as communicative acts or speech acts. Each agent $i \in M \subset N = \{1, , n\}$ calculates its best-response function u_i depending on the assumption that all local utility functions u_{ij} are

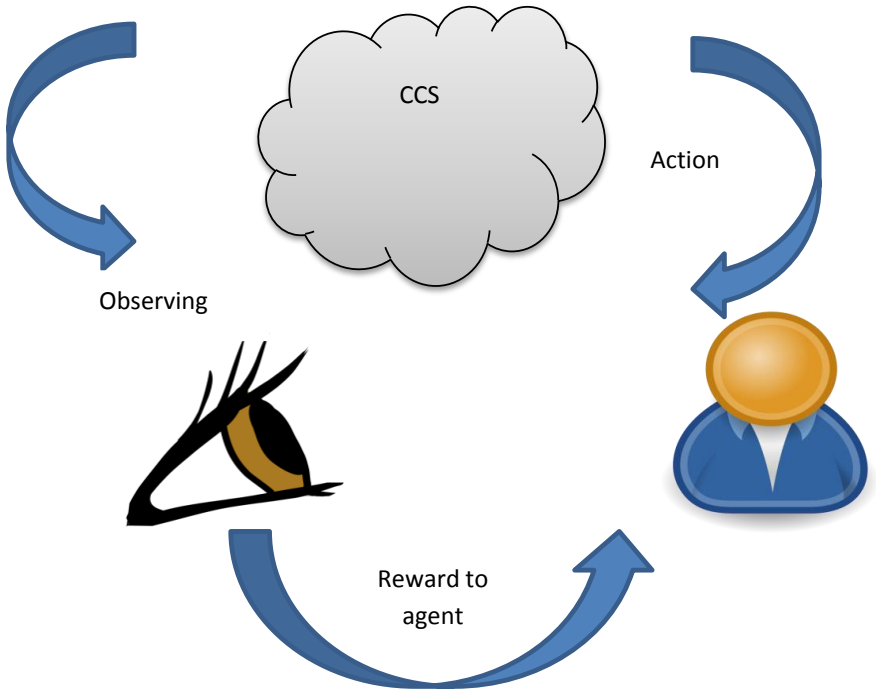

Fig.1 An agent precedes actions (information) in CCS which is deduced into a payment and a demonstration of the state which is fed back into the agent.

common information among agents. Similarly, in the backward pass, where an agent notifies the other agents of its action choice, we have supposed that the actions of the agents are well-arranged and these orderings are common information. The result of these two common information assumptions is that each agent can track the procedure in parallel, selecting an arbitrary elimination order from the set $M \subset N$. We need the following description in the sequel:

: A joint action $x^* = (x_i^*)$ is an equilibrium in dominant actions for agent $i \in M$ if we have

$$u_i(x_i, t) \leq u_i(x_i^*, t) := u_i^* \tag{3.1}$$

: for all joint actions (x_{-i}, x_i).

Proposition 3.1 For agent i satisfies (3.1) then u_i^* is unique.

Proof. Suppose there are two utility functions u_i^* and u_i^{**} for the same action x_i^* such that

$$u_i(x_i, t) \leq u_i(x_i^*, t) = u_i^*$$

and

$$u_i(x_i, t) \leq u_i(x_i^*, t) = u_i^{**},$$

which is a contradiction. □

The complete algorithm is presented as follows:

Initialize N, α, (x, t)

Select $M \subset N$ (agents with the minimum utility) with $|M| = m$

For each agent $i \in M$ in parallel

If $i \neq 1$

Wait until agent $i - 1$ sends OK.

End

Let u_{ij} be all local utility (initial and communicated) that include agent i.

Compute $\mathcal{B}_\alpha u(x, t) \equiv \sum_{n=0}^{\infty} \frac{(t^\alpha x)^n}{\Gamma(1 + \alpha n)}$.

Compute $U^*(x, t) = \max_{t \in J} \ \mathcal{B}_\alpha U(x, t)$,

Send $U^*(x, t)$ to agent $j \in M$

If $i \neq m$

Send OK to next agent $i + 1$.

Assign role j to agent i^*, (the agent who has maximum u_{ij})

Wait until all x^*_{-i} (the action of agent) are received. Note that $_{-i}$ means all agents except the agent i.

End

The OK signal is required for synchronization, ensuring an execution order of the algorithm allowing to the rejection order of the agents. Moreover, the initial distribution of utility functions of the agents in the selected set $M \subset N$ can be done as follows: agent 1 in the removal order receipts all utility functions that include this agent, agent 2 revenues all functions that contain this agent and are not distributed to agent 1, and so on, until no more utility functions are left. The capacity of the CCS does not change, because the utility function is defined in terms of it. Also, we have considered the actions $x_i, i = 1, ..., m$ of the agents $i \in M$ are ordered and these orderings are common information.

This application is preserved simultaneously by consuming the training Red Hat of hybrid IT environments. This system provides robust devices for cloud organization with private or hybrid cloud organization capabilities, active visibility technologies and advanced virtualization controlling panels.

Proposition 3.2 If the agents run the same position x_i, and it is common data in the recent state what actions the agents are considering, then the agents must be having the same utility.

Proof. The point that in the present state the action x_i of an agent i is shared data among all agents indicates that there must be a set E_S that is common dart among all agents, and in all positions of which agent I would recive action x_i, for all $i \in E$. Since E is common data among the agents, it must be self-evident to all agents, therefore, we can write $E = M$ for all i. This implies that all agents in M have take the same capacity from the CCS in the same time. But the utility function is in term of the capacity, therefore all u_i must be equal. □

4. Discussion

Applying a utility function would be informal if we had full observability of the state x. Then we could just utilize x in (2.5) and calculate the desired optimal outcome (supposing of course that we have a controllable algorithm for achieving this). Though, as we know that , x_i is exposed only to agent i. One option would be to request each agent to tell us his position x_i, but there is no promise that an agent will report his true status. Recall that each agent i forms his own favorites over outcomes, specified by his utility function $u_i(t, x_i)$ with x_i his true status. If by reporting a false type $\tilde{x}_i$ an agent i assumes to get higher payoff than by reporting x_i, then this agent may surely consider lying.

Consider the M contains $m = 5$ agents requesting one information (one item) in the same or different time $t \in J = [0, T]$. We assume a direct demonstration technique in which the agents are requested to report

their positions (x_i) during a period of time $t_i \in J$, and based on their reports, the CCS calculates an optimal outcomeu_i^* that solves (2.5). The outcome can be formulated by the following equation:

$$u(x,t) = \max_{t\in J} \sum_{i=1}^{5} u_i(x_i, t_i). \tag{4.1}$$

By using the suggested transform $\mathcal{B}_\alpha u(x,t)$, $\alpha \in (0,1]$; each agent does not depend on the report of the other agents. A simple and capable technique for computing optimal utilities in CCS, when the transit method is available is value iteration. We initialize a random utility value $U^*(x,t)$ for each status and then iteratively apply (2.5). We repeat until convergence, which is computed in a relative increase in U between two successive update steps. Value iteration converges to the optimal $U^*(x,t)$ for each state. The maximum utility appears, when the fractional power α takes its maximal value in $(0,1]$.

Table 1 shows the experimental outcomes for $M = 5$ agents and different values of the fractional order α. The utility of the CCS is estimated throughout different time $t \in [1,5]$. Obviously, the maximum utility increases, whenever increasing $\alpha \to 1$. We compere our new algorithm with the previous works.

Let the capacity of the system is equal to 1; hence the maximum value of the utility function is also equal to 1. Consider the set of the outcomes for 5-agent system

$$x_i = \{0.1, 0.15, 0.2, 0.25, 0.3\}.$$

Consequently, we obtain

$$u_i(x,t^\alpha) = \{0.1 * t^\alpha, 0.15 * t^\alpha, 0.2 * t^\alpha, 0.25 * t^\alpha, 0.3 * t^\alpha\},$$

$$\Big(t = 1, 2, ..., 5,\ \alpha \in (0,1],\ x_i \in (0,1)\Big).$$

In [1], the authors suggested the values of $\omega_j = 0.2$. In our method theses values can be determined as follows:

$$\omega_j = \frac{1}{\Gamma(1+j\alpha)}, \quad j = 1, ..., 5.$$

It is clear that the MMU is converging faster than the utility functions in [1] and [4] (see Fig. 2).

TABLE 1. Fractional multi-agent system

(α)	Time	$[1]_{\alpha=1}$	[4]	MMU
0.95	1	0.199	0.6111	0.25
	2	0.39	0.6128	0.501
	3	0.599	0.6137	0.751
	4	0.789	0.629	1
	5	0.989	0.638	-
0.75	1	-	0.656	0.338
	2	-	0.657	0.676
	3	-	0.6585	1
	4	-	0.6743	-
	5	-	0.680	-
0.5	1	-	0.6814	0.615
	2	-	0.6827	1
	3	-	0.6839	-
	4	-	0.7001	-
	5	-	0.7113	-

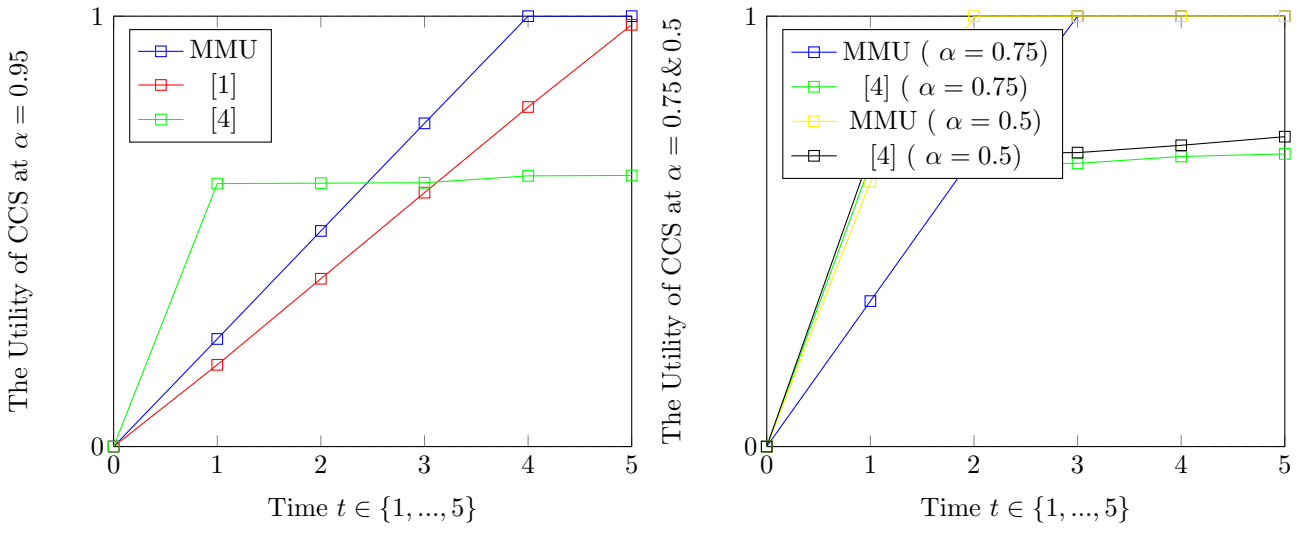

Fig.2: The utilty function for various values of $\alpha \in (0, 1]$

5. CONCLUSION

We investigated, the study of what method to maximize min utility function, a process initiated by a fractional diffeo-integral system. The method was supplementary capable and exceeds those of mathematical

software recommendation and recreating the stable profit of exchangeable with the overall cost as well as task distribution. Otherwise, the objective function is involved in the concept of the fractional function called the Mittag-Leffler sum. By utilizing this function, we familiar an adaptation to the objective function. The technique acknowledged two rewards: transforms the problem of constrained optimization into unconstrained and with an appropriate selection of the fractional order, the method is a good approximation. Moreover, one may suggest multi-connection, by applying the above method. Finally, the method can be stretched to higher-dimension, when the number of agents in the multi-agent system becomes large.

Competing Interests : The author has no competing interests in the manuscript.

References

[1] Y. K. Salih et al., A user-centric game selection model based on user preferences for the selection of the best heterogeneous wireless network, Ann. Tlcommun. 70(5-6) (2015), 1–10.

[2] Y. K. Salih et al., An intelligent selection method based on game theory in heterogeneous wireless networks, Trans. Emerg. Telecommun. Technol. 27 (12) (2016), 1641–1652.

[3] R. W. Ibrahim et al., Perturbation of fractional multi-agent systems in cloud entropy computing, Entropy 18 (1) (2016), 31.

[4] R. W. Ibrahim, A. Gani, Hybrid cloud entropy systems based on Wiener process, Kybernetes 45 (7) (2016), 1072–1083.

[5] R. W. Ibrahim, Y. K. Salih, On a fractional multi-agent cloud computing system based on the criteria of the existence of fractional differential equation, Math. Sci. (2017), 1–7.

[6] I. Podlubny, Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications. Vol. 198. Academic press 1999.

[7] A. A. Kilbas, H. M. Srivastava and J.J. Trujillo, Theory and applications of fractional differential equations. North-Holland, Mathematics Studies, Elsevier 2006.

[8] D. Baleanu, J. Machado, and A. Luo, Fractional dynamics and control. Springer Science & Business Media, 2011.

[9] R. W. Ibrahim, Fractional calculus of Multi-objective functions & Multi-agent systems. Lambert Academic Publishing, Saarbrucken, Germany 2017.

[10] R. Hilfer, Applications of Fractional Calculus in Physics. Singapore: World Scientific, 2000.

[11] N. Jafari, *et al.*, Job scheduling in the Expert Cloud based on genetic algorithms, Kybernetes, 43(8) (2014),1262-1275.

[12] M. Ashouraie, N. Navimipour, Priority-based task scheduling on heterogeneous resources in the Expert Cloud, Kybernetes: 44 (10) (2015), 1455-1471.