International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 9s, 2025
https://www.theaspd.com/ijes.php

A Genetic Algorithm for Resources Consumption Optimization within Inter-Organizational Business Process Execution

Tatiana Eyenga Ovono[1]* and Roger Atsa Etoundi[2]
[1,2] Department of Computer Science, University of Yaounde I, Yaounde, Cameroon
*[1]eyengaovono@gmail.com and [2]atsa.etoundi@ict4d.cm

Abstract: *The execution of an inter-organizational business process within a networked enterprise consumes a lot of resources such as data, throughput and time. Several researchers have made significant contribution in the field of resource allocation problems. In this paper, we address the resource consumption problem within inter-organizational business process execution. This means that, we want to show how to optimize resources utilization during the execution of multiple instances of workflow. This results in decision-making problem. It's the reason why we have decided to use genetic algorithm to overcome this issue. For this purpose, we have translated an inter-organizational workflow as a weighted directed dependency graph of web services which execute the tasks of the process. The dynamic of the workflow is observed based on the fact that for the same inter-organizational business process, we can find multiple workflow instances and, each instance is represented by the dependency graph of web services. For each dependency graph, we can derive an interaction matrix between web services. Each element of the matrix refers to the type of interactions (successful interaction, failure interaction, absent of interaction) between two services implementing the tasks of the business process. Based on this abstract representation of a workflow at runtime, we have made an analysis which have helped us to define a genetic algorithm for resources consumption optimization within inter-organizational business process execution.*

Keywords: *Dependency graph, genetic algorithm, inter-organizational business process, resource optimization.*

1. INTRODUCTION

A business process consists of a set of activities that are executed in coordination within an organizational and technical environment to reach a specific business goal [1]. A business process also refers to the well-defined logical tasks implemented by web services in order to help an enterprise to achieve their business goals. The specificity of an inter-organizational business process is that, this type of process starts in one particular enterprise and ends to another organization. Inter-organizational business processes constitute a piece of the puzzle of socio-technical system like an interoperable information system because, they need to be adaptable to the evolving organizational change. A business process can be reformed to optimize metrics like resource utilization, maximal throughput, flow time and heuristics like task resequencing, parallelism and task composition [2]. For applying all these heuristics and reaching business goals, process activities need to utilize a lot of resources such as human and non-human resources. Non-human resources can be subdivided in application resources and non-application resources.

Resource consumption aims at ensuring that each task of a particular business process uses necessary resource for being executed by a useful web service that optimize message exchange during the communication between web services despite the risks associated with the execution environment. The goal of resource consumption is to facilitate the execution of an inter-organizational business process task at runtime. Consider, for example, the online travel ticket booking process; each web service that executes a task in this process uses data, throughput and has a response time. Depending on the perturbations associated to the execution environment, the consumption of resources by this process can vary from one business process instance to another. In general, resources are not only consumed in one single process task of a single business process but in several instances of a same inter-organizational business process. That is why resource consumption optimization within inter-organizational business process execution is a decision making-problem that should consider the following aspects : the occurrence of unforeseen events during the execution of inter-organizational business process instances, the successful interaction between services implementing the business process task, the failure or the absent of interaction between web services implementing the business process task, minimize the response time to improve the performance goal of inter-organizational business process.

The goal of this paper is to formally define the problem of resource consumption optimization of inter-organizational business process after multiples instances of execution. For that, in section II, a literature review of the recent trend in the field of resource optimization in business processes is provided. In section III, we give an overview of the research method used while in section IV, we present our research contribution in details. In section V, we show the validation of our theoretical contribution through a case study. Finally, in section VI, we conclude this paper by a discussion and perspectives.

2. RELATED WORK ON RESOURCE OPTIMIZATION IN BUSINESS PROCESSES

This section presents and discusses existing survey on resource utilization in business process execution and survey on optimization model of resources. A first comprehensive study on resource utilization in business process execution was given by [3]. The authors propose a predictive model for resource utilization in decision-intensive processes. This model predicts the utilization of resources based on the order and duration of past activity executions, resources profiles and their experience. This predictive model is defined in the form of probability density functions and take into consideration the risk occurs due to the high variability that may affect operational processes in real world scenarios. By this study, the authors show that under-utilization of resources leads to higher process execution costs, and over-utilization of resources may result in process delays. The shortcomings of this model are that it's focused in human resources utilization by taking into account front office and back office employees. The work proposed by [4] on resource optimization in business processes find the optimal resource allocation schemes in processes which are modelled using color petri nets.

Regarding the survey on optimization model of resources in business process execution, the author [5] define a set of optimization problems based on some constraints such as utilization rate, waiting time and throughput. Among these problems, we have: the optimization problem constrained by utilization rate: in this case, a model of allocation of a number of resources to each role in the process, constrained by the minimal utilization rate of a specific role is defined to ensure that they utilization rate of expensive machinery are used enough and considers like the threshold. The disadvantage of this resource optimization method is that the execution of some lightweight processes may not reach the set utilization rate. To avoid this situation, it is necessary to check whether the utilization rate defined as a constraint is achievable. Another problem defined is the optimization problem constrained by waiting time [5]: for this problem, the author defines the tactical optimization of allocation of a number of resources to each role constrained by the desired maximum waiting time in the business process by taking into consideration the percentile of these waiting time distribution. When in each period a decision can be made about the number of resources hired in that period, the author defined an optimization model for allocation of the number of resources deployed in the process to each role for multiple periods. The limitations of all these resource optimization models during the execution of business processes are that, on the one hand, they focus more on the allocation of resources and do not explore aspects related to the consumption or the use of these resources once they have been allocated. In addition, the perspectives of resource optimization analysis in the execution of processes are more oriented towards human resources with notions such as role assignment and cost per resource. On the other hand, the size of the search space for solutions depends on the maximum number of resources that can be deployed in the process and the number of different roles. So that, the search space is the complete enumeration of all possible combination of the different values for the different roles. In our work, we study the problem of optimizing resource consumption in inter-organizational business process execution and we contribute to addressing the previous challenge by using the genetic algorithm, defining a data structure such as a chromosome in the form of vectors to reduce the search space and we observe an inter-organizational business process in the form of a composition of services to better highlight the communication or exchange of messages between the web services executing the tasks of the process and the interoperability criteria of different information systems. So, our model does not focus in human resources, because we analyze business process workflow by taking into consideration the web services that execute the inter-organizational business process tasks, the response time and throughput of each web service. The similar point with this model is that, we also take risks occurs into the business process execution environment.

3. RESEARCH METHOD

To conduct this research, we have used Design Science Research. As mentioned by [6], in the information system research cycle, Design Science Research contributes to create and evaluates IT artifacts intended to solve identified organizational problems. These IT artifacts can be defined as constructs, models, algorithms and instantiations. For evaluating these artifacts many types of quantitative methods can be used including optimization proofs, analytical simulation, case study and field study. For this paper, we are going to follow the Design Science Research (DSR) Process to elaborate our research contribution. This DSR process includes six steps : problem identification and motivation, definition of the objectives for a solution, design and development, demonstration, evaluation and communication [7].

4. RESEARCH CONTRIBUTION

To elaborate our scientific research contribution, we will follow intrinsically the Design Science Research Process steps. Let' us present the key concepts of the genetic algorithm that we will use in the remainder of this work.

A. Genetic Algorithm Concepts

Referring to the fact that a genetic algorithm is an algorithm inspired by the natural sciences, it is important to give a clear understanding of the concepts used like population, chromosome, individual, and gene, for solving a problem in another field different from biological science. A population is defined as a set of N individuals representing solutions to a given problem. Each individual can be represented by one or more chromosomes. A chromosome is one of the solutions to the problem, and each chromosome is made up of a set of genes. The genes correspond to the variables to be optimized or decision variables, whose values are called alleles. The general principle of genetic algorithms is to randomly select a population of individuals from a search space of solutions. This population serves as candidate solutions for optimizing the given problem. The individuals in this population are evaluated using a fitness function. Three stochastic operators are used to choose the next population, namely: selection operator, crossover operator and mutation operator. A selection mechanism is used to choose the individuals that will serve as parents for the next generation. These individuals are crossed and mutated to form an offspring. Finally, the next generation is formed by a learning mechanism combining the parent and child individuals. This procedure is repeated until a stopping condition is satisfied. Individuals are represented using an encoding scheme describes by a bit or string according to the problem domain. We distinguish several encoding schemes: real, binary, octal, hexadecimal, permutation, value-based, and tree [8]. In this paper, we are interested in real encoding schemes. A real encoding scheme helps to design genetic algorithms whose chromosomes are vectors of floating-point numbers and whose alleles are real numbers [9]. Real parameter vectors are used in our proposed genetic algorithm. Each chromosome in a population is represented by a vector. A vector consists of a set of decision variables. These decision variables are the web services implementing the tasks of an inter-organizational business process. Thus, for each instance of an inter-organizational business process execution corresponds a service utility vector at the end of the execution. After multiple business process executions, we have multiple business process instances, which also correspond to multiple service utility vectors. In the next section, we propose the definition of the resource consumption optimization problem in inter-organizational business process.

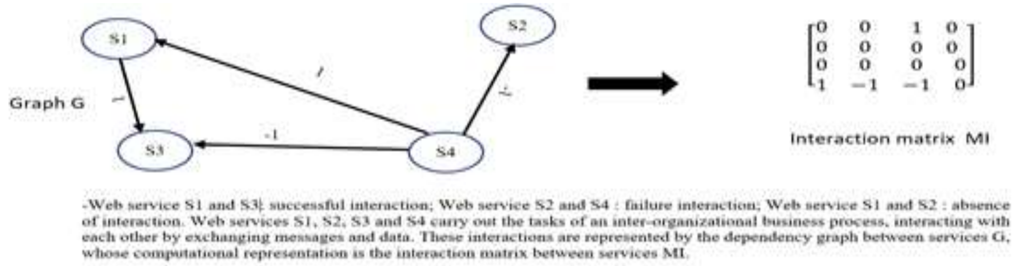

Fig. 1: A dependency graph between services and their interaction matrix.

B. Resource Consumption Optimization Problem in Inter-organizational Business Process Execution

When we decide to perceive an inter-organizational business process as a composition of the web services, the resource consumption optimization problem aims to determine the set of useful services needed to execute business process instances in networked enterprise information systems by optimizing interactions between web services and resource utilization and by taking into account the risks associated with the execution environment. The set of services executing the instances of an inter-organizational business process can be represented in the form of a dependency graph between services. Web services will be represented as nodes in a dynamic weighted directed graph. A web service is linked to other services via its ports. A web service receives messages through its input ports and sends messages through its output ports. The orientation of the arc indicates the direction of the message flow exchanged, and the weighting of the arc reflects the usefulness or otherwise of the information exchanged for the target service. Depending on the type of interaction between services when exchanging messages and resources, we distinguish three types of interaction as we can see in figure 1: those based on usable (exploitable) message exchanges, encoded by the number 1, i.e., a successful interaction; those based on non-usable exchanges, i.e., a failure, encoded by the number -1; and the absence of interaction, encoded by the number 0.

Let MI, $MI \in \{-1, 0,1\}^{\theta\times\theta}$ be the interaction matrix derived from the dependency graph between services executing business process instance tasks in interoperable networked enterprise information systems after one period of time. $MIij$ represent the type of interaction between service *i* and service *j*. Figure 1 represents this mechanism. For each interaction between two services, an unforeseen event may occur in the information system runtime environment, disrupting data transmission on either the source or target service side. These unforeseen events can include organizational contingencies (power cuts, low bandwidth), security contingencies (network attacks disrupting service operation), and organizational culture and governance structures. In this paper, we have materialized these risks using the column vector R of size θ. θ represents the order of the dependency graph between services $G = (V, E)$. An element R_i of R denotes the probability of the interruption associated with service *i*; $R_i \in [0;\ 1]$. For the sake of simplicity and in order to avoid privileging one disruption over another, the different elements of this vector will be equal to 0.5. In future studies, we will reconsider the problem by suggesting that this value is variable for each risk encountered in the information system environment.

A formal description of the problem of resource consumption in business process execution is defined as follows: let's consider θ interacting services of inter-organizational business processes of an interoperable information system of networked enterprises. Each service $j \in \{1; 2; 3; \ldots; \theta\}$ is subject to a perturbation R_j associated with the information system environment and has an interoperability cost C_j defined by its response time and throughput.

- Objects to be analyzed: interaction matrices between services MI of size $\theta * \theta$
- Question: how to determine the web services of inter-organizational business process instances with the best utility percentages U that minimize the overall cost of resource consumption $fval$?

- Solution: define an optimization model of resources consumption in business process execution as follows :

$$\begin{cases} min\, f(U)_{\ U\in\mathbb{R}^{\theta}} = \sum_{j=1}^{\theta} C_j U_j \\ s.c\ \sum_{j=1}^{\theta} MI_{ij} U_j \geq R_i\ , \qquad i \in \{1\,;2\,;3\,;\dots\,;\,\theta\},\ j \in \{1\,;2\,;3\,;\dots\,;\,\theta\} \quad (1) \\ \qquad 0 \leq U \ \leq 1 \end{cases}$$

C. Resource Consumption Optimization Model Properties

- $min\, f(U)_{\ U\in\mathbb{R}^{\theta}} = \sum_{j=1}^{\theta} C_j U_j$: means that we want to find the service utility vector U that minimizes the overall cost of resource consumption.
- $\sum_{j=1}^{\theta} MI_{ij} U_j > R_i\ , i \in \{1\,;2\,;3\,;\dots\,;\,\theta\}$: this constraint means that the disruptions induced by the information systems environment must be less than the interactions between the services of inter-organizational business processes instances.
- $0 \leq U \ \leq 1$ **:** The utility percentage of the service can be *0%* or *100%,* so the vector's components are bounded between 0 and 1.
- U : column vector, representing the service utility vector, dimension θ.
- $f\ :\ \mathbb{R}^{\theta}\ \rightarrow\ \mathbb{R}$,the objective function or the fitness function. A fitness function that takes an input X, where X is a vector with as many elements as the number of variables in the problem, calculates the function's value and returns this scalar value in its return argument Y.
- C*i* components of service interoperability cost vector. This value can be calculated using the other QoS attributes; in this paper, we only use response time and throughput.
- If U_j *∈[0;50[*, the service *j* is not useful for inter-organizational business process execution.
- If U_j *∈[50; 80[* , the service *j* is partially useful for inter-organizational business process execution.
- If U_j *∈[80;100],* the service *j* is completely useful for inter-organizational business process execution.
- The $fval$ value of the objective function, represents the overall cost of resource consumption in inter-organizational business process execution.

To calculate the components of service interoperability cost vector C using QoS attributes that emphasize exchanges between services, we will use response time (Re) and throughput (Th). *Rej* and *Thj* represent respectively the response time and the throughput of the service *j*. $Re_j\ \in [\,Re_0;\ Re_{max}]$ and $Th_j\ \in [\,Th_0;\ Th_{max}]$; Re_0 : represents the theoretical minimum response time that we aim to achieve, which improves the business process execution performance. Re_{max} : represents the theoretical maximum response time that degrades the business process execution performance. Th_0 : represents the theoretical minimum flow rate or throughput that degrades the business process execution performance. Th_{max} : represents the theoretical maximum throughput that we aim to achieve, which improves the business process execution performance.

To compensate the units of measurement between the different QoS attribute values, the values need to be normalized to lie within the interval *[0; 1]*; for example, response time should be normalized by minimization, while throughput should be normalized by maximization [10]. So, the interoperability cost C*j* of the service *j* is given by the formula:

$$C_j = \left(\frac{Vmax_{Re} - Re_j}{Vmax_{Re} - Vmin_{Re}} + \frac{Th_j - Vmin_{Th}}{Vmax_{Th} - Vmin_{Th}} \right) \quad (2)$$

$Vmax_{Re}$: represents the maximum response time of all services participating in the runtime mechanism. $Vmin_{Re}$: represents the minimum response time of all services participating in the runtime mechanism. $Vmax_{Th}$: represents the maximum throughput value of all services participating in the interoperability mechanism. $Vmin_{Th}$: represents the minimum throughput value for all services participating in the interoperability mechanism. θ : is the number of services. Re_j : represents the effective response time of service *j*. Th_j : represents the effective throughput of service *j*.

Finally, the steps taken to determine the overall cost of resource consumption in business process execution are:

- Transform the dependency graph between services into a matrix, which we have named the service interaction matrix.
- Define the risks associated with the interoperable information system environment that could disrupt exchanges between the web services implementing the business process tasks like a vector R.
- Calculate the interoperability cost associated with each service C_j (see equation (2)), taking into account response time and throughput, which are service quality attributes.
- Take into consideration the components of the service utility vector U, which are the problem variables.
- Determine the solution of equation (1), which is the extreme vector U^*, such that $f(U^*) \leq f(U)$, $\forall\ U \in \mathbb{R}^{\theta}$ So, it is the optimal solution of the problem $f(U^*)$ that gives the optimal value $fval,$ $f(U^*) = fval$, which in our work represents the overall cost of resource consumption.

To understand the interpretation of the overall cost of resource consumption which represents the best fitness value, we will refer to biological science from which genetic algorithms derive their existence. Biological science field distinguishes four types of fitness value: tautological fitness, Darwinian fitness, Thoday fitness and inclusive fitness. In each of these categories, the definition of best fitness value divides scientific opinion. However, in this paper, we take into consideration the definitions provide by Darwin's and Thoday's work.

For Darwin, fitness value is the expectation of offspring, or rather an expectation of a system, and for Thoday, fitness value is a unit of evolution [11]. On the basis of these ideas, we note that fitness value is not a probability and has no unit of measurement. This definition of the fitness value gives a sense to the results we obtained in our case study. Fitness value must therefore be understood as being relative to a particular environment at a given point time. The fitness value is only defined over one generation. The fitness value of an individual is the value of the fitness function for that individual. Thus, the best fitness value is the smallest fitness value for any individual in the population.

In the context of our work, the overall cost of resource consumption is equivalent to a mathematical definition of the best fitness value. Within the framework of mathematical definitions of the best fitness value, this value can take on a series of different values. Sometimes, fitness values extend over positive real numbers, sometimes over real numbers between 0 and 1, sometimes over real numbers greater than or equal to 1 [11]. In this paper, we consider the case where the best fitness value is greater than or equal to 1 for interpreting the overall cost resource consumption in inter-organizational business process execution, and we define the following interpretation rules:

- If the overall cost of resource consumption *fval* is less than one, then resources are under-utilized, and removing unnecessary web services in the business process optimizes resource consumption in the sense that resources can be reallocated to the remaining useful web services performing the business process tasks.
- If the overall cost of resource consumption *fval* is greater than or equal to one, then resources are over-used, and removing unnecessary web services from the business process will optimize resource consumption by the remaining useful web services.

D. A Genetic Algorithm for Resource Consumption Optimization

When we decide to use a genetic algorithm as a linear program that would run inside of an optimizer, it's important to know what we want the genetic algorithm to do. To answer this question, the literature review offers us two points of view on the use of genetic algorithms: for design problems and for repetitive problems. As mentioned by the authors [12] , in design problems, the genetic algorithm is used to develop an optimizer to find the right solution. In this case, we have many trials, and it's not a worry if some executions end up with bad results. In repetitive problems, the genetic algorithm is an optimizer in its own right, and in this case, we aim to reduce the probability of obtaining bad results by carrying out a single execution very often. In this scenario, the algorithm can then be used later in different problem instances to compare different versions of this algorithm experimentally, taking into account their performance measures. In this paper, we use genetic algorithm for design science problem.

1) Code scheme, fitness function and stochastic operators: Real coding scheme is used in the proposed genetic algorithm. Each chromosome in the population is represented like the service utility vector $U_{t_i} = (U_{t_{i,1}}, U_{t_{i,2}}, \dots, U_{t_{i,\theta}})$, where θ represent the number of web services implementing the inter-organizational business tasks. $U_{t_{i,j}} \in [0\%; 100\%[= [0; 1[\ ; j \in [1; \theta]$ refers to the utility percentage of the service *j*. So that U_{t_i} is a vector where each component represents the utility percentage of a service when the ith-instance of the inter-organizational business process is executed. For example, the service with identifier θ -1 in the U_{t_1} vector is said to be useful for the interoperability of business process instances. Figure 2 represents an example of solution.

With regard to the initial population, initial solutions are randomly generated according to the encoding rules of solutions, that is, the service utility vector is chosen randomly for each inter-organizational business process instance. A service utility vector is made up of a set of decision variables.

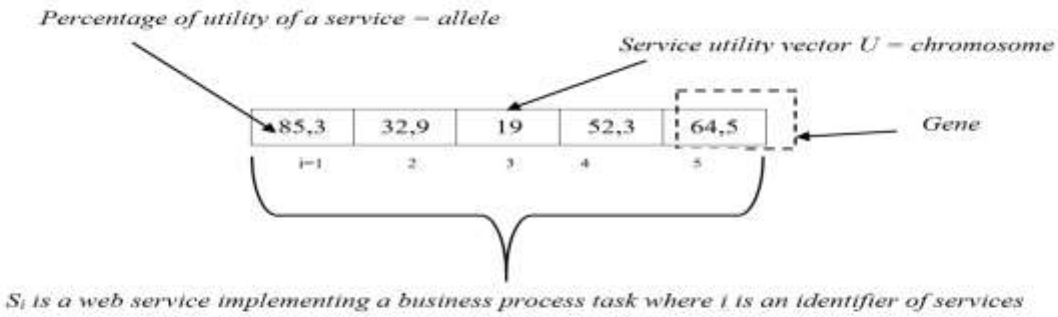

Fig. 2: A chromosome

These decision variables are the web services implementing the tasks of an inter-organizational business process. So, for each instance of an inter-organizational business process execution corresponds a service utility vector at the end of the execution. After multiple business process executions, we have multiple business process instances, which also correspond to multiple service utility vectors. In the case of the initial population, these service utility vectors are derived from service interaction matrices. The variability of the interaction matrix is due to the different screenshots made on the interacting information system at given instants $t_1, t_2, \dots, t_n$; these screenshots represent an instant of execution of the business process studied. This means that at one instant t_1, we capture exchanges between the services of a business process under study, and then at another instant t_n, we capture yet more exchanges and we obtain different interaction matrices. In other words, at one instant t_1 of the execution of inter-organizational business, we can obtain the interaction matrix MI_{t_1}, and at another instant t_n, we can obtain another interaction matrix MI_{t_n}. Based on these ideas the fitness function is defined as follows:

$$Fitness(U_{t_i}) = f(U_{t_i}) + R * \sum \langle g_i(U_{t_i}) \rangle \quad (3)$$

R is the penalty factor and f is the objective function. By defining our fitness function using penalties, the idea of this method is to transform a constrained optimization problem into an unconstrained one by adding (or subtracting) a certain value to the objective function depending on the degree of constraint violation present in a certain solution. More information about penalty functions can be found in [12]. Regarding the stochastic operators, for this study, we will use the tournament selection operator, polynomial mutation operator and the SBX (Simulated Binary Crossover) operator defines by [13].

2) Elite-based learning mechanism and termination condition: The principle of elitism states that an individual's probability of reproduction is proportional to its relative fitness. Elitism refers to the fact that the best individuals must always participate in reproduction [14]. The learning strategy therefore enables the genetic algorithm to define a movement for exploring the best solutions in the search space. In our case, the learning strategy consists of reserving utility vectors for services where at least half of the genes have alleles with a percentage utility of the service greater than 50%. The termination condition of algorithm is defined when the number of generations is reached.

3) Steps of the genetic algorithm for resource consumption optimization in business process execution:

- Initialization: generate an initial population whose size corresponds to the number of execution instances of the business process under study, given that θ is the number of services used by this process.
- Calculate the value of the fitness function for each service utility vector of the current generation and retain the best vectors.
- Select the parent vectors using the stochastic tournament selection operator.
- Modify the current generation by applying the SBX crossover operator with probability *pc* to the vectors.
- Modify the generation by applying the mutation operator with probability *pm* to the vectors.
- Use the learning mechanism to obtain the best utility vectors for the services of the current generation.
- Replace the vectors reserved in step (b.) with the best vectors obtained after the elitism mechanism to create a new generation.
- Count the number of generations *Ng*. If *Ng* reaches the predefined threshold *Ngmax*, the algorithm ends, otherwise, go to step (b.).

4) A proposed approach for resource consumption optimization in inter-organizational business process execution: The resource consumption optimization approach helps us to know how we can use the predefined algorithm in a real networked enterprise environment. It serves as a transition between the defined theoretical model and its practical application. This approach is based on the following steps and encompassed the table of correspondence between genetic algorithm concepts and resource consumption concepts in business process.

- Model the inter-organizational business process studied using BPEL4WS language.
- Identify the algorithm inputs: the services and number of services executing the tasks of an instance of the business process studied; the response time (minimum and maximum) and throughput (minimum and maximum) intervals relative to the execution of all the process tasks.
- Execute the algorithm to obtain the overall cost of resource consumption of the business process.
- Browse the components of the solution vector to identify useful web services, and interpret the resulting global cost of resource consumption.

Table 1: Correspondence between genetic algorithm concepts and resource concepts

Genetic algorithm concepts	Resource consumption concepts in business process execution
Population size	Number of times a business process is executed (number of instances)
Objective function	Resource consumption optimization model
Chromosome	Service utility vector
Gene position	Service identifier
Population	Set of service utility vectors
Allele	Percentage of service utility
Number of genes	Number of services executing the tasks of an inter-organizational business process
Number of generations	Program stop condition

5. CASE STUDY

The aim of this case study is to optimize the overall cost of resource consumption in an enterprise network by analyzing the inter-organizational business process of cash withdrawal. The goal is to identify the set of unnecessary services that consume the most resources and obscure the execution of several process instances. For that, we present the case study in details, apply algorithm, analyze the results obtained, and verify the complexity properties of the proposed algorithm.

A. Presentation of the case study

The case study described here presents the collaboration and interoperability between the GIMAC (Interbank Electronic Payment Group of Central Africa) information system and the information systems of two partner banks, A and B. GIMAC is a banking ecosystem made up of a group of financial institutions, banks, mobile money operators, and aggregators present in all the countries of the CEMAC (Central African Economic and Monetary Community) sub-region. GIMAC's main mission is to implement full interoperability and interbanking for CEMAC electronic payment systems. With its GIMACPAY digital service, electronic financial transactions are federated (federated approach to interoperability), facilitating the transfer of money from one mobile account to another mobile account (or bank account) of another operator, and vice versa.

In this study, we focus on the inter-organizational business process of withdrawing money from GIMAC ATMs. This business process involves the interoperation of three different information systems: Bank A's information system, Bank B's information system, GIMAC's information system, and an X customer. Data collection was based on an interview with GIMAC's mobile financial services engineer concerning the GIMACPAY solution (which federates all electronic payment systems and means in CEMAC) and the inter-organizational business process studied. Information from the GIMAC website was also used to complete our analysis. During a period of immersion within the company, we worked with the expert to learn more about the business process under study.

*1) **Relationships Between the Companies Studied:*** Customer X needs to withdraw money from his bank account A (Bank A's information system) at Bank B's ATM (Bank B's information system) using his GIMACPAY card (GIMAC's information system). Although all the contextual elements (e.g. the amount to be withdrawn is less than the account amount) are correct, the transaction failed. For this reason, we are interested in studying the inter-organizational business process of cash withdrawal. To find out more, let's define the process scenario.

*2) **Process Scenario:*** The process of withdrawing cash from a GIMAC ATM consists of two sub-processes: the process of obtaining cash and the process of monitoring the transaction. The process of obtaining cash starts when customer X inserts his card into an authorized Bank B ATM. The ATM checks the card's compliance with the GIMAC system, and also verifies the customer's balance. If the card complies and the required amount is less than the current balance, the ATM debits the account. It is then up to the customer to retrieve their card and take their cash. If this is not the case, no transaction is carried out, and the customer simply gets his card back. In either case, the transaction is recorded in the interoperable information system. The transaction tracking process is as follows: based on the transaction carried out by the customer, a transaction report is drawn up for each customer making a withdrawal, according to date (time, month, year), bank (issuing bank and receiving bank) and location (country, city) where the transactions were carried out. When a withdrawal fails, the bank information system associated with the ATM receives notification of the failed transaction; it checks and identifies the origin of the card, then transmits a report (an automatic transaction) to the GIMAC information system, which notifies the bank owning the customer account.

B. Modelization of the inter-organizational business process studied

We have used BPNM notation to model the inter-organizational business process of cash withdrawal. Knowing that we are interested in the execution of that business processes and that we cannot access to the real GIMAC execution environment due to the sensitivity of the financial data contained in GIMAC, we will simulate the cash withdrawal process using a business process simulator called "Bizagi Modeler". The aim of this simulation was to gain insight into the minimum and maximum response times, and the minimum and maximum throughput associated with each service executing a task of the business process instances studied. At the end, one hundred instances of the business process studied were simulated in order to identify the maximum and minimum response times, bearing in mind that the minimum and maximum throughput was considered as a resource provided by an Internet service provider. Figure 3 shows the process studied and the various simulations linked to resource consumption in this process. After simulating 100 instances of the cash withdrawal process, we deduce that the minimum response time is 11 min 54 s and the maximum response time is 54 min 58 s. The minimum throughput is 2 Mbps and the maximum is 622 Mbps.

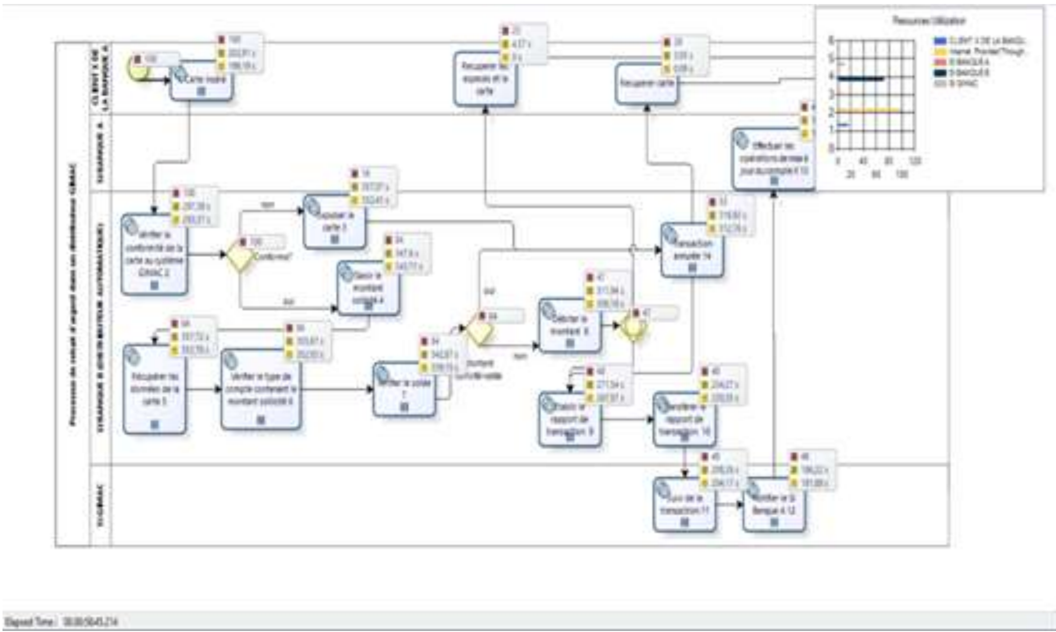

Fig. 3: Simulation of the cash withdrawal process using different resources

C. Identification of genetic algorithm inputs

Table 2: Proposed genetic algorithm inputs

Parameters	Value
Population size	100
Number of generations	150
Number of real variables or number of services	14
Probability of crossover *Pc*	0.9

Probability of mutation *Pm*	0.5
Number of Run experiments	5
Exponent for polynomial crossover operator *n*	2
Exponent for polynomial mutation operator ηm	100
Static penalty factor *R*	10

Fig. 4: Results of the fourth experiment

D. *Execution of the proposed genetic algorithm for resource consumption optimization in the inter-organizational business process of cash withdrawal studied*

The characteristics of the machine we used are as follows: Intel(R) Core(TM) i5-8265U CPU @ 1.80 GHz, C language and gcc compiler (MinGW.org GCC-6.3.0-1) 6.3.0 to implement the proposed genetic algorithm and perform the tests related to our case study. After five experiments, we notice that the fourth experiment (see figure 4) produces better results. These results are shown in Table 3. Analysis of these results reveals that services 4, 10 and 11 are not useful for the execution of the cash withdrawal process studied,and the overall cost of resource consumption is: -125.934474.

Table 3: Results of resource consumption optimization of the cash withdrawal business process studied

Service i	Service Utility Percentage
1	0.84
2	0.92
3	0.96
4	0.41
5	0.83
6	0.61
7	0.91
8	0.99
9	0.99
10	0.32
11	0.31
12	0.89
13	1.00
14	1.00

6. DISCUSSION

In the case of the inter-organizational cash withdrawal business process studied, the overall cost of resource consumption is equal to -125.934474 < 1; we conclude that resources are under-utilized, and removing unnecessary web services with id 4, 10 and 11 in the business process, optimizes resource consumption in the sense that resources can be reallocated to the remaining useful web services performing the business process tasks. Figure 5 shows that this value is the best solution found by the proposed algorithm after eighteen generations. The complexity of the algorithm is of the order of O(gnm) with g the number of generations, n the population size and m the individual size. We therefore have g=150, n=100 and m=14.

7. CONCLUSION

In this article, the problem we address the problem of resource consumption optimization in inter-organizational business process execution. To solve this problem, we have generated new knowledge in our field of information systems by implementing a genetic algorithm for resource consumption optimization in inter-organizational workflow. These contributions were validated in practice within a network of companies called GIMAC, through the study of the inter-organizational business process of cash withdrawal in a GIMAC-approved distributor. The results show that the overall cost of resource consumption is negative, which prove that in this business process, resource consumption can be optimize in the sense that resources can be reallocated to the remaining useful web services performing the business process tasks. Another decision-making we can derive in this case study is that the interoperable information system of this business network will no longer be scalable after 18 generations. As a continuation of this work, we plan to reconsider certain parts of the genetic algorithm program in order to apply parallelism to the execution of tasks in-memory; test the proposed algorithm by implementing other crossover and mutation operators; add other quality-of-service attributes such as reliability to calculate the service cost interoperability and conduct another study where the values of risk vector is different to 0.5.

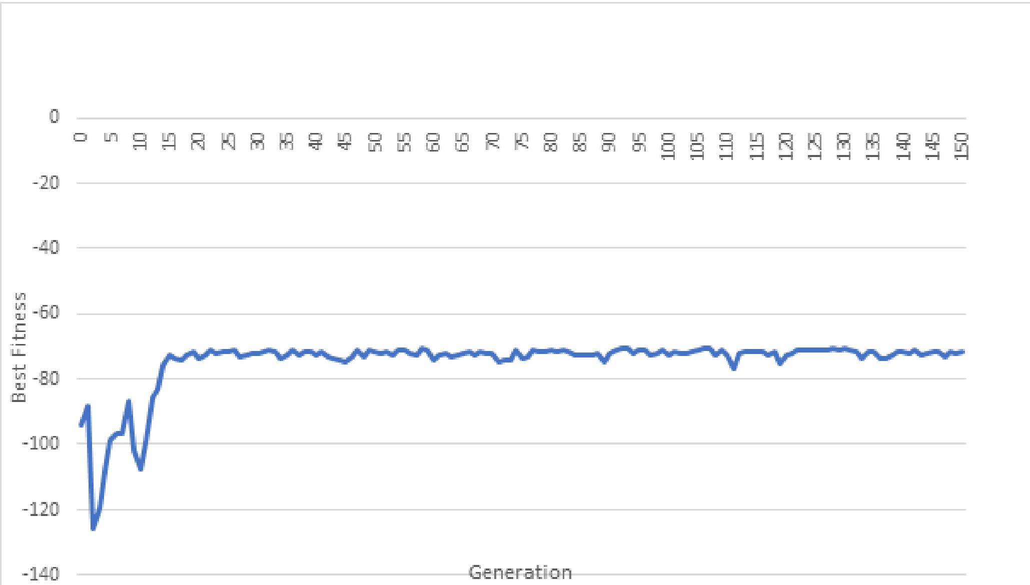

Fig.5. Proposed Genetic Algorithm Convergence Analysis

REFERENCES

[1] L. Pufahl, F. Stiehle, S. Ihde, M. Weske, et I. Weber, « Resource allocation in business process executions—A systematic literature study », *Information Systems*, vol. 132, p. 102541, mars 2025, doi: 10.1016/j.is.2025.102541.

[2] G. Tsakalidis et K. Vergidis, « Business Process Redesign: A Systematic Review of Evaluation Approaches Prior to Implementation », *Decision Making: Applications in Management and Engineering, Volume 7, Issue 1 (2024) 79-98*, 2024.

[3] S. Sperl, G. Havur, S. Steyskal, C. Cabanillas, A. Polleres, et A. Haselbock, « Resource Utilization Prediction in Decision-Intensive Business Processes », *the Austrian Research Promotion Agency*, p. 128-144, 2024.

[4] Y.-W. Si, V.-I. Chan, M. Dumas, et D. Zhang, « A Petri Nets based Generic Genetic Algorithm framework for resource optimization in business processes », *Simulation Modelling Practice and Theory*, vol. 86, p. 72-101, août 2018, doi: 10.1016/j.simpat.2018.05.004.

[5] Peters, S. P. F., . Analysis and Optimization of Resources in Business Processes. [Phd Thesis 1 (Research TU/e / Graduation TU/e), Industrial Engineering and Innovation Sciences]. Technische Universiteit Eindhoven., 2021.

[6] A. R. Hevner et al., « Design Science In Information Systems Research », *MIS Quarterly*, vol. Vol. 28 No. 1, p. 75-105, 2004.

[7] J. vom Brocke, A. Hevner, et A. Maedche, « Introduction to Design Science Research », in *Design Science Research. Cases*, J. vom Brocke, A. Hevner, et A. Maedche, Éd., in Progress in IS. , Cham: Springer International Publishing, 2020, p. 1-13. doi: 10.1007/978-3-030-46781-4_1.

[8] S. Katoch, S. S. Chauhan, et V. Kumar, « A review on genetic algorithm: past, present, and future », *Multimed Tools Appl*, vol. 80, n° 5, p. 8091-8126, févr. 2021, doi: 10.1007/s11042-020-10139-6.

[9] A. H. Wright, « Genetic Algorithms for Real Parameter Optimization », in *Foundations of Genetic Algorithms*, vol. 1, Elsevier, 1991, p. 205-218. doi: 10.1016/B978-0-08-050684-5.50016-1.

[10] Agushaka J. O. et al., « Effect of Weighting Scheme to QoS Properties in Web Service Discovery ». (IJCSIS) International Journal of Computer Science and Information Security, Vol. 7, No. 3, March 2010, 2010. [En ligne]. Disponible sur: (IJCSIS) International Journal of Computer Science and Information Security, Vol. 7, No. 3, March 2010

[11] S. Kimbrough, « The concepts of fitness and selection in evolutionary biology », *Journal of Social and Biological Systems*, vol. 3, n° 2, p. 149-170, avr. 1980, doi: 10.1016/0140-1750(80)90007-X.

[12] A. E. Eiben et J. E. Smith, *Introduction to Evolutionary Computing*. in Natural Computing Series. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015. doi: 10.1007/978-3-662-44874-8.

[13] K. Deb et R. Bhushan Agrawal, « Simulated Binary Crossover for Continuous Search Space », *Complex Systems*, vol. 9, p. 115-148, 1995.

[14] N. Saini, « Review of Selection Methods in Genetic Algorithms », *International Journal Of Engineering And Computer Science ISSN:2319- Index Copernicus value: 58.10, 76.25. DOI: 10.18535/ijecs/v6i12.04*, vol. 6, n° 22261-22263, 2017.