2016 XI International Symposium on Telecommunications (BIHTEL)
October 24-26, 2016, Sarajevo, Bosnia and Herzegovina

Optimization of business processes by automatic reallocation of resources using the genetic algorithm

Almir Djedović, Emir Žunić
Info Studio Ltd, Sarajevo
Sarajevo, Bosnia and Herzegovina
almir.djedovic@infostudio.ba
emir.zunic@infostudio.ba

Zikrija Avdagić, Almir Karabegović,
Faculty of Electrical Engineering, University of Sarajevo
Sarajevo, Bosnia and Herzegovina
zavdagic@etf.unsa.ba
akarabegovic@etf.unsa.ba

***Abstract*— Business process management is the process of modifying or adjusting an organization's business process in order to achieve higher productivity or lower costs. Each company or organization has a value creating process that usually involves people, machines and information. One of the main problems with such processes is that it is very difficult to predict how much of each resource is actually needed. In light of the above, the objective of this paper is to implement a methodology that is capable of optimizing the allocation of resources to tasks in a given business process. In this paper, the genetic algorithm was used for optimization. The idea is that once the units are properly presented, the optimal schedule of users should be determined using the genetic algorithm. The fitness function includes Key Performance Indicators of process: waiting time and cost of the resource. Since al the users are not qualified in performing all the tasks in the process, the algorithm has to consider minimal and the maximal available number of users for each activity. The usability of this approach is tested in the process of credit requirement. Finally, the results are compared to the current work process.**

Keywords—Business process management, Process optimization, Resource allocation, Business Process Modeling Genetic algorithm, Simulation

I. Introduction

Business processes present set of logically connected activities using the company resources, with the aim of satisfying the customer's needs for products or services of appropriate quality in an adequate period of time, with realizing some value at the same time [1]. During the global competition in the market, companies can achieve competitive advantage only by offering cheaper and better products and services, and for realizing such objectives, effective and innovative business processes are needed.

Business Process Management (BPM) is a process-oriented management discipline [2]. Business Process Management is a top-down methodology designed to organize, manage, analyze, and reengineer the processes used in an organization. BPM is defined as "supporting business processes using methods, techniques and software to design, enact, control and analyze operational processes involving humans, organizations, applications, documents and other sources of information" [3]. This means that BPM is linked with areas of Artificial Intelligence (AI) as Knowledge Management (KM), Knowledge Engineering (KE) and Business Intelligence (BI).

Nowadays, the optimization of business processes is an important element of existing and functioning of any organization. Continuous improvement of business processes results in reducing the costs, time saving, improved quality and competitive advantage as well as staying in the market. Processes are of crucial importance to be effectively implemented for the organization.

There are multiple approaches of optimizing a business process. One approach is to redesign the process model so that the tasks are done in different order. The purpose of such optimization is to reduce the amount of tasks that get held up by process bottlenecks. Relocating a bottleneck task to a position which is reached by less process instances should improve overall performance. Another situation where relocating a task to a later stage might be useful is when one has a very parallel business process and some tasks need all of the branches to be finished before it can be executed. Leaving such tasks until the end of the process might have a significant impact on process throughput.

Another approach for optimizing business processes is to leave the process model as it is but try out different amounts of resources, consider training the employees so that the task execution times are reduced or rearrange the resources which are responsible for executing a given task. One can always assume that increasing the amount of resources will improve the average cycle time of the process until the minimum cycle time is reached [4]. Meanwhile, there are a lot of cost considerations that have to be taken into account when deciding to train one's employees so that they would perform their tasks more quickly. If the improvements in cycle time are not large enough, they will not justify the training cost. Rearranging tasks between resources is another optimization technique. In this technique one task can be allocated from one resource to another. This might be useful when the training costs are low and the utilization of one resource is very low. However, caution should be taken when reassigning tasks as it is not easy to predict whether the new assignees will manage to handle the additional workload or not. Another downside of

978-1-5090-2902-0/16/$31.00 ©2016 IEEE

such rearrangement is that the quality might suffer if people from one role have to start doing the job of other specialists.

This paper is about the optimatization of business processes by reallocation of resources. It is very difficult to manually check which the best schedule of the users in the process is. It is quite evident in complex business processes and/ or in the process with a large number of participants in the process. In this paper, genetic algorithm is used for solving this problem. With the help of genetic algorithm, it is necessary to find such a schedule of the participants in the process in order to maximize or minimize the corresponding fitness function.

The work is organized as following: First, the related work is discussed in more details in Section II. In Section III, there is the principle of genetic algorithm. Optimatization of business processes by reallocation of resources realized by using genetic algorithm is discussed in Section IV. A concrete implementation is given in Section V in order to demonstrate the applicability of the approach. Section VI provides the conclusion of the paper.

II. Related Work

The idea of using genetic algorithm in optimization of business processes is not new. The genetic algorithm in the optimization of business processes, used by other authors is described in the following section.

The paper [5] examines the various metaheuristic searching algorithms in problems of reallocation of resources and arrangement. There are three algorithms being compared: genetic algorithm, the algorithm of simulated annealing and tabu search. The problems of reallocation of resources, with or without constraints have been analyzed. In problems of reallocation of resources without constraints, even the data from the algorithm of ant colonies from the second paper have been taken into consideration, and later, all four algorithms have been compared. The genetic algorithm resulted in the best performances, considering the least number of iterations with its best optimal solution found in them. The same was in problems of reallocation of resources with constraints. Genetic algorithm found the best optimal solution in the shortest period of time.

In paper [6], the authors deal with the problem of optimal reallocation of human resources considering practical constraints that may occur during the development of the project. All constraints are being considered through the coefficient in the fitness function. Those coefficients are: cost, continual efficiency being reflected through multitasking, coefficient of balance of allocated tasks and coefficient, which shows how successful the distribution of tasks among participants in the process is. After using genetic algorithm, results show that it is necessary to perform multitasking in a shorter period of time as possible, to minimize performing of tasks without defining priorities in the process, and it is necessary to allocate a larger number of resources when it is possible. These results were presented to programmers and managers as well, who consider the attained results useful to software managers.

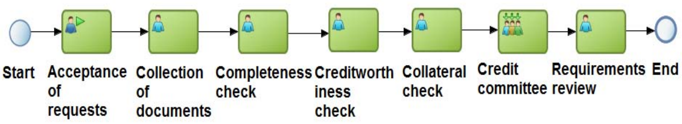

Figure 1. Process of credit requirement management

Considering a long period of time necessary for calculations the main disadvantage of genetic algorithm, the paper [7] compares performances of genetic algorithm with a hybrid genetic algorithm. Algorithms were compared in terms of duration of the project and the time of algorithm operation. Results show that integrated (hybrid) genetic algorithm has an approximate optimal time of project execution and gives better results than a standard genetic algorithm. Integrated genetic algorithms are estimated to give more realistic and promising solutions than genetic ones.

Optimization of business processes in cloud architecture can be implemented in a way where the genetic algorithm could be used for the optimal selection of web services that can realize the required service. D. Agrawal in paper [8] uses a genetic algorithm in order to select a set of services that meet agreed quality of services. The fitness function is equal to the total service quality. Chromosomes were composed out of requirements and services that can realize the requirement. The method of roulette selection was used for the selection. This algorithm was suggested to be implemented in the load balancing.

Unlike paper [6], the methodology of optimization of business processes will be presented in this one. Simulation model implemented in MATLAB will be used in order to ensure as accurate results as possible. The usability of this approach will be demonstrated in the process of credit requirement.

III. Genetic Algorithm for Business Process Optimisation

Genetic algorithm is guided random search algorithm based on the principles of evolution and natural genetics. It combines the exploitation of the previous results with the exploration of new areas of the search space. In a genetic algorithm, a population of strings (called chromosomes or the genotype of the genome), which encodes candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem, evolves toward better solutions. Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible [9]. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, fitness is calculated for each individual, then, more individuals are being selected out of a current population (based on their fitness) and modified (by recombing or mutation) into a new population. New population is used in the new algorithm iteration. The algorithm usually ends when the maximal number of generations is created or when satisfactory fitness is reached. If algorithm ends by creating a maximal number of generations, then fitness can or cannot be satisfied [10].

Chromosomes, often referred to as a genome, represent the set of parameters defining the solution of the problem genetic algorithm is trying to solve. The chromosome is often represented as a simple string, although a wide variety of other data structures are also used [11]. The genetic representation of individuals is called a genotype. The main criteria in selecting the string representation of the search node are that the new string generated from the application of genetic operator must represent legal search node for the problem.

The reproduction process is typically based on the fitness values of the strings. The principal is that string with higher fitness value will have higher chance of surviving to the next generation.

In genetic algorithms, crossover is a genetic operator used to vary the programming of a chromosome or chromosomes from one generation to the next. It is analogous to reproduction and biological crossover, upon which genetic algorithms are based. A crossover is a process of taking more than one parent solutions and producing a child's solution from them. There are methods for selection of the chromosome.

In genetic algorithms of computing, mutation is a genetic operator used to maintain genetic diversity from one generation of a population of algorithm chromosomes to the next. It is similar to biological mutation of the individuals. Mutation changes one or more genes in the chromosomes. In mutation, the solution may change entirely from the previous solution. Hence GA can come to a better solution by using mutation. Mutation occurs during evolution according to a user-definable mutation probability. This probability should be set low. If it is set too high, the search will turn into a primitive random search [12]. One of the criteria of genetic algorithm convergence is the change of the fitness function from one to another generation. If the fitness function doesn't change that is an indication that genetic algorithm has found an optimal solution. In order to escape local optima mutation operator can be used.

IV. Automatic Reallocation of the Resources by Using Genetic Algorithm

Human resource management usually called the personnel management, in an inevitable and essential function of the modern management. Human resource management is a particular scientific discipline. Intellectual capital and large internal reserves are in human resources and their activation can achieve significant results regarding success and initiative. Human resource management is a very demanding job in business processes. It is very difficult to determine the optimal number of resources in the process as well as their schedule. Concerning less or unqualified resources, bottlenecks occur in the very execution of the process. These bottlenecks prolong the execution of the process, reduce the effectiveness of the process and cause discontent among users. One of the possible solutions of this problem is adding new users. Let us consider the process of credit requirement in banks in Figure 1. For the presentation of this process, the worldwide standard for modeling business processes BPMN 2.0 was used [13].

Table 1. Simulation Scenarios for process of opening bank account to legal entities

Scenario	**#1**	**#2**	**#3**
Acceptance of the requests	3	3	3
Collection of documents	2	3	2
Completeness check	2	2	3
Creditworthiness check	1	1	2
Collateral check	2	2	2
Credit committee	3	3	3
Requirements review	1	1	1
Total cost	**67679**	**60618**	**47277**

Generally, there are several ways for acquiring process model. A process model can be attained by analyzing log files containing information about the execution of the process, by using the documentation containing detailed process description, or by interview with the owner of the process. An interview with the owner of the process was conducted in this paper and the model of the process was presented in the Figure 1. Oracle Business Process Management [14] was used as software for implementation. The process begins with receiving the credit requirement (Acceptance of the requests). After filling the given form, the necessary documents are collected (Collection of documents) and verified if they are complete (Completeness check). When all the documents are competed, the next step is verification of the creditworthiness of the applicant of the requirement (*Creditworthiness check*) and collateral check (*Collateral check*). If everything is successfully done, the credit committee (*Credit committee*) decides on the credit requirement and verifies requests that could be accepted or rejected. The final activity is Requirements review. The process is modelled according to the work principle of processing of credit requirement in one of the banks in Bosnia and Herzegovina. Interviewing the owner of the process, it was found out that 3 participants performed the step Acceptance of the request, 2 participants performed Collection of documents etc. A detailed schedule of users is shown in the first column of the Table 1. There is also other information for the process of credit requirement obtained from the interview with the owner in order to create a valid simulation model. This is information about the duration of certain tasks in the process, the number of submitted requests etc. After setting the parameters of the business process model, the simulation has been initiated [14]. The last row in the table shows the total cost of process execution obtained by simulation. The total cost of the process depends on the number and cost of resources, process execution time, bottlenecks in the process etc. For the schedule of the users presented in Scenario 1, the total process execution cost is 67679 BAM, which can be seen in Table 1. Increasing the number of users on the activities Collection of documents from 2 to 3, the cost of the process almost remained unchanged: 60618 BAM (Scenario 2). On the other hand, if the number of users is increased on the activities Completness check from 2 to 3 and Creditworthiness check from 1 to 2, the total cost of the process is reduced to about 2000 BAM.

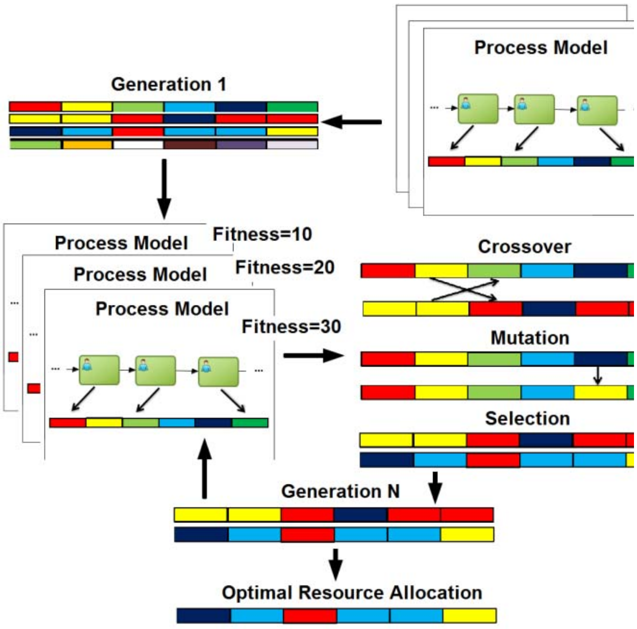

Figure 2. Model for Business Process Optimization using Genetic Algorithm

This shows that the process of human resource management in business process is demanding and complex. It is difficult to determine the optimal schedule of the users in the process of minimizing the total execution cost or some other criterion. Increasing the number of users in the process can make the very process better or worse. Improvements are seen in Scenario 2 and 3. It is unacceptable to manually change the users' schedule. In fact, that approach is the time required and it is particularly expressed in complex processes with numerous numbers of activities and users. Therefore, a genetic algorithm for searching the optimal solution is suggested in this paper. The idea is that genetic algorithm, through iterative searching of the space of possible solutions determines the schedule of users which will minimize or maximize a defined fitness function. It is important to pay attention that all the users being available in the process are not qualified to perform all the tasks in the process. There is a constraint of the total number of users that could be involved in the process, as well as a constraint of qualified users for each activity in the process. It is important to notice that this constraint exists in terms of the maximum and minimum number of users in every activity. There are some activities that can occur in the process where the number of users cannot be less than a specific value,

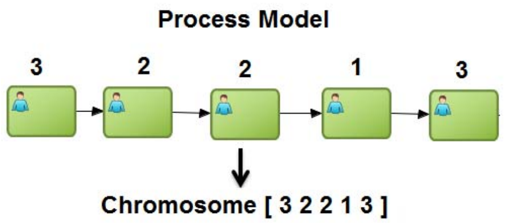

Figure 3. Model for Business Process Optimization using Genetic Algorithm

```
Generate population P0
Initialize population P0 and generation counter g=1;
while (g < the maximum number of generation) {
        Run the simulation based on chromosome;
        Evaluate fitness of each chromosome;
        Select c1 and c2 from population using
        tournament selection;
        Mutate c1 and c2;
        Generate c3 from c1 and c2;
        Replace the weakest candidate of Pg with c3
        g= g+1;
end;
```

Algorithm 1: Pseudocode of Genetic Algorithm

i.e. deciding committee and it has 3 users. It would not be correct to take that kind of schedule of users, where the number of users would be less than 3 on the committee. Therefore, the resource pool is defined in the process. A resource pool is a collection of resources that can perform a given role or that have a certain capability required to perform certain tasks. The mathematical model of the problem being solved is shown in Figure 2. and pseudocode in Algorithm 1.

A. Representation

Every schedule of users in the process presents one chromosome. Length of chromosomes matches the number of activities in the process. If it is supposed there are 5 activities done by 3, 2, 2, 1 and 3 users, respectively. The chromosome that is equivalent to that model of the process is of the form [3 2 2 1 3], as shown in Figure 3.

B. Fitness Function

After, the way of representing is defined the process model via chromosomes, it is necessary to define the fitness function. In the beginning of the optimization of business processes, it is preferable to choose the fitness function with KPI (Key Performance Indicators). These indicators can be an average time of the process, the total process cost, resource cost, an average waiting time in the entire process or specific (critical) activities. In addition, all the constraints that may occur in the process can be taken as penalties in the fitness function. For example, an unauthorized schedule of users can provide a great value of fitness function in the process of its minimization. In this way, solutions will be rejected during the selection due to a high value of the fitness function.

C. Selection

Tournament selection is probably the most popular selection method in genetic algorithm due to its efficiency and simple implementation [15]. In tournament selection, *n* individuals are randomly selected from the wider population, and then, all the individuals are compared (compete) with each other. The individual with the highest fitness wins and will be included as one of the next generation population. The number of individuals competing in each tournament is referred to as tournament size, commonly set to 2 (also called binary tournament). Tournament selection also gives a chance for all individuals to be selected and thus it preserves diversity, although keeping diversity may degrade the convergence speed.

D. Crossover

A single point crossover will be used for crossover. Such crossover operator is selected as the most efficient in task allocation in the process in [16]. Out of all compared crossover operations, single point crossover managed to find the most satisfactory solution in the shortest period of time. Single point chooses a random integer *n* between 1 and the size of the chromosome, and selects the vector entries numbered less than or equal to *n* from the first parent, selects genes numbered greater than n from the second parent, and concatenates these entries to form the child. For example if parents are: $p_1 = [1\ 2\ 3\ 4\ 5]$, $p_2 = [2\ 2\ 1\ 1\ 2]$ and *n=3* then the child is $c = [1\ 2\ 3\ 1\ 2]$.

E. Mutation

In Mutation, the Adaptive Mean Mutation Operator (AMMO) will be used. AMMO generates directions that are adaptive with respect to the last successful or unsuccessful generation. AMMO uses a combination of the Gaussian normal and Cauchy distributions, but rather than creating a different offspring with each the density. Functions are convolved into one distribution and only one offspring is created. For AMMO, offspring was created [17]:

$$\vec{x}_i' = \vec{x}_i + \vec{\sigma}_{1i}N(0,1) + \vec{\sigma}_{2i}C(0,1) \tag{1}$$
$$\vec{\sigma}_{1i}' = \vec{\sigma}_{1i}\exp\left(\tau' N_i(0,1) + \tau N(0,1)\right) \tag{2}$$
$$\vec{\sigma}_{2i}' = \vec{\sigma}_{2i}\exp\left(\tau' N_i(0,1) + \tau N(0,1)\right) \tag{3}$$

Where the $\vec{x} \in \mathbb{R}^n$ is the vector of the population. $N(0,1)$ is a Gaussian normally distributed random variable with mean 0 and $\sigma^2 = 1$ newly generated for every σ of every population member. Also, $N_i(0,1)$ is also a Gaussian normally distributed random variable with mean 0 and variance $\sigma^2 = 1$ but is only generated only once for each population member. The values τ and τ' allow for the evolution of the strategy parameters of the EA and are typically set, by Schwefel's suggestion [18]:

$$\tau \propto \left(\sqrt{2\sqrt{n}}\right)^{-1} \tag{4}$$
$$\tau' \propto \left(\sqrt{2n}\right)^{-1} \tag{5}$$

The AMMO method was selected, because it converges faster in most of the time together with FMO (Fast Evolutionary Programming) and MMO (Mean Mutation Operator) methods. More about AMMO method and the reason for its selection can be found in [17]. The newly created individual is added to the generation and the weakest one is removed from it.

V. Case Study

The presented method of the optimization of business processes using genetic algorithm, was tested in the process of credit requirement (Figure 1.). MATLAB was used for the implementation. SimEvents library [19] was used for the

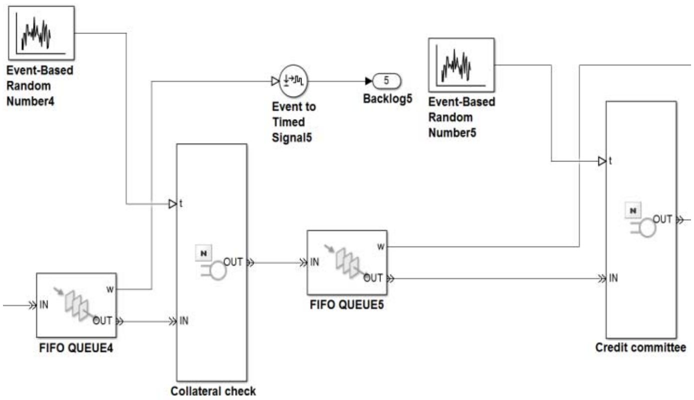

Figure 4. Equivalent process model in Matlab

in MATLAB by using SimEvents library. The equivalent presentation of the process. The first step was to present the process from BPMN 2.0 notation with the appropriate model model of the process is shown in Figure 4. As previously mentioned, for adjusting the process model parameters as the execution time of some tasks in the process, generating time of new process instances, minimal number of users necessary for performing the task etc., information were provided by the owner of the process. Also, the limits of certain resources have been defined for each activity in the process, i.e. the capacity of the resource pool. They are presented in the form of vector constraints:

$$l_b = [1\ 1\ 1\ 1\ 1\ 3\ 1] \tag{6}$$
$$u_b = [3\ 4\ 3\ 5\ 4\ 3\ 3] \tag{7}$$

Which means there cannot be less than 1 or more than 3 users on the first activity, there cannot be less than 1 or more than 4 users on the second activity, there cannot be less than 1 or more than 3 users on the third activity etc. It is necessary to point that Credit committee always has to have exactly 3 members involved. In the processes where the clients are being served, such as this one, it is necessary to ensure less waiting time for the clients. Therefore, the fitness function has to take into consideration the sum of all waitings times in the process. The schedule of the users in the process of such a model is obvious to be equal to vector u_b, because when the maximal number of users is scheduled, then the clients will wait for the service as short as possible. The cost of the process has to be considered, too. The fitness function includes 2 KPI's: total waiting time and the resource cost. The vector of the resource cost was defined per one hour of work for each activity in the process:

$$cost = [10\ 12\ 12\ 12\ 10\ 20\ 10] \tag{8}$$

All delays in the process have to be punished by penalties in the fitness function. The minute of waiting of client in the process is taken to be equal to the cost of working hour of the cheapest activity. In accordance with all these facts mentioned, the fitness function has the following form:

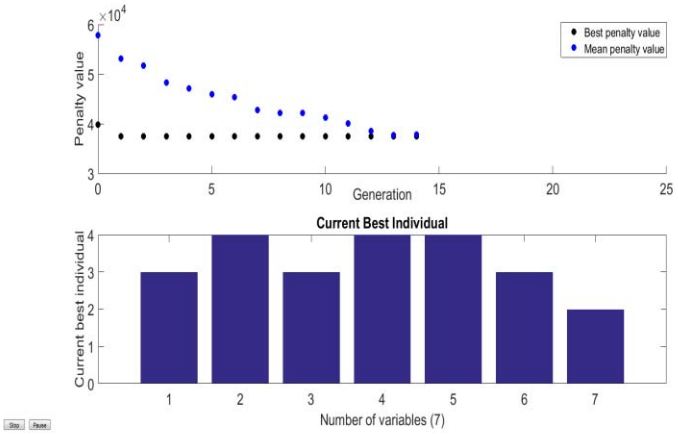

Figure 5. Results after the application of genetic algorithm

$$fitness = waiting_{time} \cdot 10 + resource_{number} \cdot cost' \quad (9)$$

where the $waiting_{time}$ is the total waiting time of the instances (clients) in the process and equals to the sum of the waiting time in all activities, while $resource_{number}$ is the number of scheduled resources in the process. After application of genetic algorithm, results are obtained and shown in Figure 5. Process has been tested for the period of one month. The upper part of the Figure 5. shows that the algorithm has managed to converge to the best solution after about 15 generations. Genetic algorithm converged because there is no significant improvement in the values of fitness function of the population from one generation to the next. The change in fitness function was below the threshold value, which was 1e-6. Also, the case study was relatively small and it was proved by brute-force approach that the solution obtained by genetic algorithm was optimal. More complex processes and processes with more participants are expected to have a higher number of generations. The lower part of Figure 5. presents the algorithm that gave the optimal resource schedule:

$$resource_{number} = [3\ 4\ 3\ 4\ 4\ 3\ 2]$$

The total delay in the process for the optimal model is 3719.6 minutes. The total resource cost is 282, while the value of the fitness function is 37478. It is interesting to compare the results with the current condition used in the bank. It is important to mention that the current number of users in the process has been determined on the basis of experience and many years of work as shown in Table 1. in the third column (Scenario 3). Total delay obtained for this model is 4227 minutes. Total resource cost is 204 and value of the fitness function 42474. It is obvious the resource cost has been increased, but the total waiting in the process has drastically been reduced. This result is logical because every additional resource in the process results in increasing the total cost of the resource as well as speeding up the process. Comparison of current and optimizes process are shown in Table 2. It is necessary to mention that the real total cost of the optimal process is higher than it is shown in the table, because here only resource costs are taken into account. Actually, in realistic processes, for each new job, it is necessary to provide the space, equipment etc., which represents additional expenses. However, additional costs are expected to be during a longer period of time, compensated with retaining the clients, preserving their confidentiality and acquiring new clients. It should be noted that the algorithm was able to find the optimal solution considering practical constraints that can occur in the process, such as, on one hand, the minimal number of users that can perform a certain activity, and on the other hand, the maximal number of the users which cannot be exceeded because of the insufficient number of available positions or some other reasons. The algorithm has found the solution taking into account the number of users on Credit committee that has to be 3. There are always 3 members of the committee who decide if the credit requirement will be accepted or rejected. The algorithm was run several times on the same process. Suggested mutation was used in order not to fall into a local minimum. Also, the proposed algorithm was performed on larger problems. Business processes with 9, 14 and 20 activities were tested and the algorithm has found the solution. Due to complexity of business process algorithm needed more time and more generation were needed to generate to find the optimal resource reallocation. For example, in business process with 20 activities algorithm has found the solution in 93 generation.

Table 2. Comparison of current and optimized process

KPI	Current process	Optimized process
Waiting time (min)	543.2	5.5
Total resource cost	3000	3350

VI. Conclusion

In this paper, method for optimization of business processes using the genetic algorithm is presented. First, it is necessary to define the model of business process and corresponding parameters in the process such as: time of the activity duration, maximal and minimal number of resources and number of generated instances of the process. This information is provided by the owner of the process. Furthermore, genetic algorithm was used for obtaining the optimal schedule of the users in the process. The fitness function takes into consideration KPI processes such as waiting time of process instances and the total cost of the resources. It is proved the algorithm is able to find the optimal schedule of users from the aspect of defined fitness function. Also, the algorithm takes into consideration the practical constraints such as the limited number of qualified resources or fixed number of resources as it was in activity Credit Committee. The algorithm has already found an optimal solution for a small number of generations, which is better in state of fitness function compared to the current schedule of users in the process.

REFERENCES

[1] Smith, Howard; Fingar, Peter: Business Process Management. The Third Wave. Meghan-Kiffer, US, 2003.

[2] R. K. L. Ko, "A computer scientist's introductory guide to business process management (BPM)", in *ACM Crossroads*, vol. 15(4), Summer 2009, pp. 11–18.

[3] W. M. P. van der Aalst, A. H. M. ter Hofstede, and M. Weske, "Business process management: A survey" in *Proc. Business Process Management: International Conference, BPM 2003*, Eindhoven, the Netherlands, ser. LNCS. Springer, June 26-27, 2003, pp. 1–12.

[4] J. Xu, C. Liu, X. Zhao, "Resource Allocation vs. Business Process Improvement: How They Impact on Each Other" in *BPM,* vol. 5240, 2008, pp. 228-243.

[5] A. M. Connor, A. Shah, "Resource Allocation Using Metaheuristic Search", in Proceedings of the 4th International Conference on Computer Science and Information Technology, 2014, pp. 353-364.

[6] J. Park, D. Seo, G. Hong, D. Shin, J. Hwa, D. H. Bae, "Human Resource Allocation in Software Project with Practical Considerations", in *International Journal of Software Engineering and Knowledge Engineering,* vol. 25(1), 2015, pp. 5-26.

[7] J. L. Kim, "Integrated Genetic Algorithm and its Applications for Construction Resource Optimization", in Proceedings of the 2010 Winter Simulation Conference, 2010, pp. 3212-3219.

[8] D. Agrawal, H. L. Jaiswal, I. Singh, K. Chandrasekaran, "An Evolutionary Approach to Optimizing Cloud Services", in *Computer Engineering and Intelligent Systems*, vol. 3(4), 2012, pp. 47-54.

[9] D. Beasley, D. R. Bull, R. R. Martin, "An Overview of Genetic Algorithms,Part 1, Fundamentals", in *University Computing*, vol. 15 (2), pp. 58-69

[10] W. Banzhaf, P. Nordin, R. E. Keller, F. D. Francone, "Genetic Programming – An Introduction", Morgan Kaufmann Publishers,US, 1998.

[11] J. Yang, C. Wu, H. P. Lee, Y. Liang, "Solving traveling salesman problems using generalized chromosme genetic algorithm", in *Progress in Natural Science,* vol. 18, 2008, pp. 887-892.

[12] M. Obitko (2016, June, 13). *Crossover and Mutation*. [Online]. Available: http://www.obitko.com/tutorials/geneticalgorithms/crossover-mutation.php

[13] Object Management Group, (2016, June, 14). Business Process Management and Notation. [Online]. Available: http://www.bpmn.org/

[14] Oracle Corporation. Business Process Composer User's Guide for Oracle Business Process Management. U.S., 2013.

[15] D. E. Goldberg, K. Deb, "A Comparative Analysis of Selection Shemes Used in Genetic Algorithms", in *G. J. E. Rawlins (Ed.), Foundations of Genetic algorithms,* 1991, pp. 69-93.

[16] J. Magalhães-Mendes, "A Comparative Study of Crossover Operators for Genetic Algorithms to Solve the Job Shop Scheduling Problem", in *WSEAS Transactions on Computers,* vol. 12(4), 2013, pp. 164-173.

[17] A. C. Koenig, "A Study of Mutation Methods for Evolutionary Algorithms", in *Advanced Topics in Artifical Intelligence,* 2002, pp. 1-8.

[18] T. Bäck, H-P. Schwefel, "An Overview of Evolutionary Algorithms for Parameter Optimization", in Evolutionary Computation, vol. 1(1), 1993, pp. 1-23.

[19] MathWorks, (2016, June, 18). SimEvents – Model and simulate discrete-event systems. [Online]. Available: http://www.mathworks.com/products/simevent.