

Implementación de un algoritmo genético en JavaScript para solucionar el problema de asignación de docentes para un periodo académico en el IST Yavirac

Implementation of a genetic algorithm in JavaScript to solve the problem of assigning teachers by academic semester at the Yavirac Higher Technological Institute

Robayo, Pablo

Pablo Robayo
pablorobayo@gmail.com
Universidad Internacional de la Rioja, Ecuador

Ecuadorian Science Journal
GDEON, Ecuador
ISSN-e: 2602-8077
Periodicidad: Semestral
vol. 5, núm. Esp.3, 2021
esj@gdeon.org

Recepción: 25 Septiembre 2021
Aprobación: 04 Octubre 2021

URL: <http://portal.amelica.org/ameli/jatsRepo/606/6062738022/index.html>

DOI: <https://doi.org/10.46480/esj.5.3.159>

Los autores mantienen los derechos sobre los artículos y por tanto son libres de compartir, copiar, distribuir, ejecutar y comunicar públicamente la obra sus sitios web personales o en depósitos institucionales, después de su publicación en esta revista, siempre y cuando proporcionen información bibliográfica que acredite su publicación en esta revista. Licencia de Creative Commons Las obras están bajo una <https://creativecommons.org/licenses/by-nc-nd/4.0/deed.es>



Esta obra está bajo una Licencia Creative Commons Atribución-NoComercial-SinDerivar 4.0 Internacional.

Como citar : Robayo, P. (2021). Implementación de un algoritmo genético en JavaScript para solucionar el problema de asignación de docentes para un periodo académico en el IST Yavirac. Ecuadorian Science Journal, 5(3), 262-271. DOI: <https://journals.gdeon.org/index.php/esj/article/view/159>



Resumen: La asignación de los docentes que van a dictar una asignatura en un periodo académico es un problema común en las instituciones de educación superior. Este problema ya ha sido resuelto con diferentes técnicas, Yo presento una solución de una aplicación web con aplicación de un algoritmo genético en el backend. El IST Yavirac, al no contar con personal administrativo, realiza esta actividad con su planta docente, que deben combinar sus actividades de docencia con esta actividad administrativa. El problema de asignación de docentes por semestre, debido al número de variables y su complejidad es de tipo NP-completo, es decir no existe una única solución, sino más bien existe un conjunto de soluciones válidos. Para resolver este problema, se optó por el uso de Inteligencia Artificial, específicamente con el diseño e implementación de un algoritmo genético; esta solución se implementa sobre una web API REST en un servidor Web con NodeJS y puede ser implementado en la nube para producción. Este artículo, muestra el en-foque utilizado para dar una solución práctica, apoyado en el uso de ontologías para representar los objetos de negocio en cromosoma en lugar de la representación clásica que se basa en cadenas de caracteres. Este enfoque permite el cálculo de soluciones viables, que puedan ser aceptadas por los usuarios; su cálculo se reduce a unos pocos minutos en comparación con la solución manual que puede llegar a tomar hasta cinco días de trabajo.

Palabras clave: Inteligencia artificial, Procesamiento de datos, Lenguajes de computación, Algoritmos.

Abstract: The allocation of teachers who are going to teach a subject in an academic period is a common problem in higher education institutions. This problem has already been solved using different techniques. I am presenting a solution of a web application applying a genetic algorithm in the backend. IST Yavirac has no administrative staff, and carries out this activity with its teaching staff, who must combine their teaching activities with this administrative activity. Due to the number of variables and their complexity, the problem of assigning teachers per semester is NP-complete; that is, there is no single solution, but rather a set of valid solutions. Artificial Intelligence was chosen to solve this problem, specifically through the design

and implementation of a genetic algorithm. This solution is implemented on a REST API web on a Web server with NodeJS and can be implemented in the cloud for production. This article shows the approach used to provide a practical solution, supported by ontologies to represent business objects on chromosomes, instead of the classical representation based on character strings. This approach allows viable solutions to be calculated, which can be accepted by the users. The calculation takes only a few minutes compared to the manual solution that can require up to five working days.

Keywords: Artificial intelligence, Data processing, Computer languages, Algorithms.

INTRODUCTION

Yavirac Higher Technological Institute

The Ecuadorian Higher Technological Institutes (ISTs) are institutions of higher education, which grant higher technological third level degrees ("Organic Law of Higher Education, LOES," 2020). IST Yavirac is a public higher education institution under the Secretariat for Higher Education, Science, Technology and Innovation (SENESCYT). ("Yavirac," n.d.).

To teach the classes, presently online due to the Covid 19 pandemic, IST Yavirac offers courses in the different careers in three sessions: morning, afternoon and evening; At the beginning of each semester, the vice-rector and the career coordinators carry out an academics analysis and planning process in order to meet the academic needs of that period; this includes factors such as: number of enrolled students by career, number of classes per day and per subject, availability of teachers by time and day, experience of teachers in different subject areas, administrative role, number of hours for each subject, hours to be assigned to each teacher, type of contract, availability to teach one or more careers.

There are some restrictions to consider, such as:

- A teacher can teach classes on two consecutive days,
- Each teacher has a minimum and maximum number of hours to teach.
- A teacher has areas of specialization and cannot teach subjects outside of that area.

The assignment of teachers by subject for an academic period is done manually. Over several days of the week each career identifies the classes to be taught and based on their previous experience, teachers are randomly assigned. This also leads to competition for teachers, and overloads some and leaves others without enough hours. It takes one and two weeks to complete the process for each of the six careers. At IST Yavirac, the product of the planning is the teacher distribution and their names. The teacher distribution is recorded in a spreadsheet. Due to the number of variables and the manual nature of the process, it requires considerable time, and due to cost, we end up using the first solution obtained, which is not necessarily the best possible solution.

Based on the teacher distribution, reports are generated specifying the subjects a teacher has to teach by career and day, a report of the total teaching hours assigned by career and a consolidated report of the total hours per teacher.

Nature of the problem

Based on the characteristics we reviewed in detail, we realized that the problem of assigning teachers by subject belongs to the broad category of schedule assignment problems. Scheduling problems are NP-complete (Nuntasen and Innet, 2007); thus, for each case, these problems have restrictions and resources which, when combined, determine the effectiveness of the result obtained (Beligiannis et al., 2009).

This problem was addressed previously, and several optimization methods were used to find near optimal solutions. Some of the methods for solving this problem are:

- Local search techniques,
- Reasoning based on cases and heuristics,
- Artificial intelligence techniques such as: genetic algorithms, evolutionary algorithms, Taboo search, ant colony systems and metaheuristics.

Artificial intelligence and genetic algorithms

Artificial intelligence is the study and science of making intelligent systems to obtain feasible solutions to different problems; genetic algorithms are part of this field of knowledge (Ansari and Bakar, 2014).

Genetic algorithms are based on the evolutionary theory postulated by Charles Darwin: randomness in variation, together with the law of natural selection, constitutes a technique for solving problems with a wide spectrum of application. ("Algoritmos Genéticos - Fernando Sancho Caparrini," 2019). His concept was developed by John Holland at the beginning of the 1970s (Tsoukalas et al., 1997).

A genetic algorithm is a probabilistic search algorithm based on the mechanics of the principles of natural selection and natural genetics. Genetic programming allows the creation of evolutionary algorithms to map data to a result when there is no established formula. (Kumar et al., 2010). These algorithms use character structures, and emulate living beings that evolve over time according to the principle of survival of the fittest, using an information exchange scheme that is structured randomly. New structures are created in each generation using parts of the fittest members of the old group. (Beligiannis et al., 2009).

Development tools

Javascript

Javascript is a programming language that was released on December 4, 1995. It was created by Netscape (Netscape Communications Corporation, 1995).

Some of the main features of Javascript are:

- It is an interpreted or compiled programming language.
- It is multiparadigm
- It runs on a single thread
- It supports declarative, imperative, and object-oriented programming.
- It can be used for frontend applications.
- As well as web server applications with NodeJS
- In databases: CouchDB

("JavaScript | MDN," 2021)

To date, the massive development of web applications has transformed JavaScript into a ubiquitous tool, and applications can be made for:

1. Frontend: AngularJS, Vue, React,
2. Back end: NodeJS, Deno,
3. Mobile apps for Android - iOS: react native
4. Desktop Applications: Electron

(Mohan, 2020)

NodeJS

NodeJS is an open source program, which allows the execution of Javascript on the server side. It is built on Google's "V8" software. The program allows long server processes to be executed with excellent performance. (Tilkov and Vinoski, 2010).

NodeJS is installed in more than 100 countries and is used in: web application development, business applications, big data, embedded systems, etc. It has many advantages: it is designed to optimize performance and scalability in web applications and APIs, which is ideal for executing Artificial Intelligence processes; it includes an NPM package manager and can run on various operating systems such as Microsoft Windows, OS X, Linux, Solaris, FreeBSD, and others ("Express/Node introduction - Learn web development | MDN," 2021)

A web API is an interface that allows access to the resources of a web server through HTTP and the exchange of data using the JSON format (Wittern et al., 2017)

MATERIAL AND METHODS/METHODOLOGY

Next, I will present the implementation of the prototype of the genetical algorithm I constructed to solve the problem of assigning teachers per semester at the Yavirac Higher Technological Institute; in business terms, the resulting product is known as distributive.

We are going to implement a genetic algorithm written in JavaScript that consists of the following functions shown in Table 1:

TABLE 1.
Genetic algorithm functions

Function	Description
mutationFunction	Mutation function
crossOverFunction	Crossover function
fitnessFunction	Fitness function
createFirstPhenotype	Create the starting population

Author

Although the scope of this document is focused on the implementation of the genetic algorithm on the server, I will briefly explain in a general way the general architecture of the solution.

Solution architecture

I designed a web application prototype using an open source stack that uses React for the frontend; an API is implemented to allow accessing the genetic algorithm in a NodeJS server and the information is saved in a MySQL database. It can be seen in Figure 1

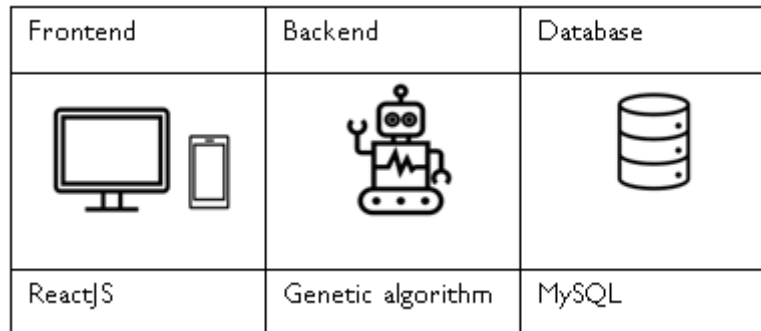


FIGURE 1.

Figure
Author

Creation of the initial population

In the construction of the prototype to solve the problem of assigning teachers by semester at IST Yavirac, data from the Software Development Career of a previous semester consisting of 59 classes and 22 teachers were considered.

Each class has the following attributes:

- Number of hours of duration.
- It belongs to an area of knowledge, for example: humanistic, programming, mathematics.
- It belongs to a session: morning, evening, night

Each teacher is represented in the system by the following characteristics:

- Areas of expertise (array)
- Minimum number of teaching hours to be assigned.
- Actual assigned teaching hours.
- Subjects to be taught by academic period (array)
- Total teaching hours assigned

Some of the restrictions that must be met for it to be a valid assignment are:

- Each teacher can teach consecutive sessions: morning-afternoon, afternoon-evening.
- Each teacher should only teach subjects that are within their areas of expertise

Each professor must be assigned the number of hours assigned for the academic period. In a genetic algorithm, the population is like a set of chromosomes; it is common to model a chromosome as a binary representation. This need not be the case as it can be a symbolic representation, a chain of things that determine how the system behaves (MIT OpenCourseWare, 2014). Ontologies allow a representation of a particular domain to represent reality and identify artifacts and the relationships between them (Athiththan et al., 2018).

So, using ontologies, I determined that for this problem one class constitutes the chromosome; and it consists of two objects: subject and teacher.

TABLE 1.
Interpretation of the objects of the genetic algorithm in the context of the problem to be solved.

Genetic Algorithm Object	Interpretation
Chromosome	Represents a specific class, for example: Id: 1, Career: marketing, Session: morning, teacher: Juan Flores, subject: statistics
Population	Teacher distribution.

Author

Upon reviewing the literature, I found that the representation of chromosomes usually involves a string of characters: "11011"; this makes a lot of sense since the algorithm was implemented by applying a structured programming paradigm.

Class 1					Class 2					...	Class n				
1	1	0	1	1	2	1	2	3	4	...	0	1	1	4	1

FIGURE 2
Representation of a chromosome of a class that uses a numerical representation with 5 positions: course, teacher, student section, classroom and period.
Matias et al., 2018

To solve this problem, my proposal uses ontologies, and a chromosome is an object composed of a subject and a teacher. In JavaScript you can create objects that materialize the objects identified in the ontologies.

Clase 1		Clase 2		Clase 3	
					
Asignatura	Docente	Asignatura	Docente	Asignatura	Docente

FIGURE 3
Representation of a chromosome of a class that uses a numerical representation with 5 positions: course, teacher, student section, classroom and period.
Author

```
{
  id: 36,
  clase: 'DS3MB',
  asignatura: 'CÁLCULO DIFERENCIAL E INTEGRAL',
  horas: 4,
  area: 'MATEMATICAS',
},
```

FIGURE 4
Subject object in JavaScript.
Author

```
{
  id: 11,
  docente: 'MAURICIO PRIETO',
  areas: ['PROGRAMACION', 'BASEDEDATOS'],
  horasMin: 20,
  horasAsignadas: 0,
  asignaturas: [],
  asignaturasCount: 0,
  investigacionFormativa: 0,
  investigacionAcademica: 0,
  vinculacion: 0,
  gestion: 0,
  totalHoras: 0,
},
```

FIGURE 5
Teacher object in JavaScript.
Author

The createFirstPhenotype function assigns each subject to a random teacher from a filtered set. To create the filtered set, the function searches for all teachers whose areas of expertise coincide with the area of knowledge to which the subject belongs. In this way, we try to ensure that even for the generation of the first population, it is the most optimal. This also reduces processing times.

Distributive evaluation (fitnessFunction)

The function runs through the phenotype and on each chromosome, it assesses:

- The total hours assigned to the teacher versus the minimum load of hours that they have defined.
- It checks which subjects correspond to the teacher's areas of expertise.
- The function ends by assigning a score.

In this version, the function does not evaluate cases in which the maximum number of defined teaching hours is exceeded. In that case, users later make a manual correction.

Mutation (mutationFunction)

The population size is fixed for this particular problem. We then specified that what must always change are the teachers but not the subjects. The function carries out the following steps:

1. Get a random gene (teacher).
2. Within the chromosome, identify the area of knowledge to which your subject belongs.
3. Get another random chromosome
4. Perform gene swapping (teachers)

Reproduction (crossOverFunction)

In this operation, the genetic variation of the chromosome is generated towards the next generation; It is implemented by calculating a random cut-off point and then chromosomes are copied between subjects.

```
{
  id: 11,
  docente: 'MAURICIO PRIETO',
  areas: ['PROGRAMACION', 'BASEDEDATOS'],
  horasMin: 20,
  horasAsignadas: 0,
  asignaturas: [],
  asignaturasCount: 0,
  investigacionFormativa: 0,
  investigacionAcademica: 0,
  vinculacion: 0,
  gestion: 0,
  totalHoras: 0,
},
```

FIGURE 6:
Code of the crossover function implemented in JavaScript.
Author

RESULTS

The algorithm was configured to execute ten thousand evolutions; the execution took approximately 5 minutes. The execution time of the genetic algorithm may vary depending on the characteristics of the server where it is executed.

```
createFirstPhenotype();
var geneticAlgorithmConstructor = require('../node_modules/geneticalgorithm/index.js');
var geneticAlgorithm = geneticAlgorithmConstructor({
  mutationFunction: mutationFunction,
  crossoverFunction: crossoverFunction,
  fitnessFunction: fitnessFunction,
  population: [primerDistributivo],
});

console.log("Starting with:");
console.log(primerDistributivo);

for (var i = 0; i < 10000; i++) geneticAlgorithm.evolve();
var best = geneticAlgorithm.best();
delete best.score;
console.log("*****");
console.log("Finished with:");
console.log(best);
```

FIGURE 7
Code of the genetic algorithm constructor function.
Author

In this prototype, the output of the algorithm is shown in JSON data format.


```

asignatura:
{ id: 1,
  asignatura: 'Fundamentos de programación',
  horas: 6,
  area: 'programación' },
docente:
{ id: 2,
  docente: 'Lorena Ch',
  areas: [ 'programación' ],
  horasMin: 23,
  horasAsignadas: 0,
  asignaturas: [ ] },
asignatura:
{ id: 2,
  asignatura: 'Análisis de Sistemas',
  horas: 5,
  area: 'programación' },
docente:
{ id: 3,
  docente: 'Fanny S',
  areas: [ 'programación' ],
  horasMin: 23,
  horasAsignadas: 0,
  asignaturas: [ ] },
asignatura:
{ id: 3,
  asignatura: 'Introducción al desarrollo de software',
  horas: 4,
  area: 'programación' },
docente:
{ id: 3,
  docente: 'Fanny S',
  areas: [ 'programación' ],
  horasMin: 23,
  horasAsignadas: 0,
  asignaturas: [ ] },

```

FIGURE 8

Example of JSON file produced by the execution of the program on de NodeJS server.

Author

Discussion

Preliminary results show results close to those obtained when assigning manually. Currently the work takes about 40 hours, but with automation it is possible to have the first version of the distribution in minutes; later, fine-tuning and adjustments can be carried out, which will notably reduce the time and effort required by IST Yavirac to develop a viable distribution. In this specific problem, the objective function corresponds to a set of chromosomes of fixed size.

CONCLUSIONS

In this contribution, it designed, developed, and applied a genetic algorithm to solve the problem of assigning teachers by subject in an academic period for the Yavirac Higher Technological Institute and created feasible distributions to help teachers have reasonable class schedules.

Using ontologies, it was possible to accurately model business objects and identify the relationships they have with other objects. I think the different functions can be fine-tuned to better include and define compliance with the different constraints of the problem. Future work will redefine the assessment used in the Fitness function to include more business rules and restrictions. Rules should also be included to pre-assign subjects to certain teachers as previously determined by the authorities.

BIBLIOGRAPHIC REFERENCES

- Algoritmos Genéticos - Fernando Sancho Caparrini [WWW Document], 2019. URL <http://www.cs.us.es/~fsancho/?e=65> (accessed 8.31.21).
- Ansari, A., Bakar, A.A., 2014. A Comparative Study of Three Artificial Intelligence Techniques: Genetic Algorithm, Neural Network, and Fuzzy Logic, on Scheduling Problem, in: 2014 4th International Conference on Artificial Intelligence with Applications in Engineering and Technology. Presented at the 2014 Artificial Intelligence with

- Applications in Engineering and Technology (ICAIET), IEEE, Kota Kinabalu, Malaysia, pp. 31–36. <https://doi.org/10.1109/ICAIET.2014.15>
- Athiththan, K., Rovinsan, S., Sathveegan, S., Gunasekaran, N., Gunawardena, K.S.A.W., Kasthurirathna, D., 2018. An Ontology-based Approach to Automate the Software Development Process, in: 2018 IEEE International Conference on Information and Automation for Sustainability (ICIAfS). Presented at the 2018 IEEE International Conference on Information and Automation for Sustainability (ICIAfS), IEEE, Colombo, Sri Lanka, pp. 1–6. <https://doi.org/10.1109/ICIAFS.2018.8913339>
- Beligiannis, G.N., Moschopoulos, C., Likothanassis, S.D., 2009. A genetic algorithm approach to school timetabling. *Journal of the Operational Research Society* 60, 23–42. <https://doi.org/10.1057/palgrave.jors.2602525>
- Express/Node introduction - Learn web development | MDN [WWW Document], 2021. URL https://developer.mozilla.org/es/docs/Learn/Server-side/Express_Nodejs/Introduction (accessed 8.29.21).
- JavaScript | MDN [WWW Document], 2021. URL <https://developer.mozilla.org/es/docs/Web/JavaScript> (accessed 8.27.21).
- Kumar, M., Husain, M., Upreti, N., Gupta, D., 2010. Genetic Algorithm: Review and Application. *SSRN Journal*. <https://doi.org/10.2139/ssrn.3529843>
- Ley Orgánica de educación superior, LOES, 2020.
- Marias, J.B., Fajardo, A.C., Medina, R.P., 2018. A Hybrid Genetic Algorithm for Course Scheduling and Teaching Workload Management, in: 2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM). Presented at the 2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), IEEE, Baguio City, Philippines, pp. 1–6. <https://doi.org/10.1109/HNICEM.2018.8666332>
- MIT OpenCourseWare, 2014. 13. Learning: Genetic Algorithms.
- Mohan, M., 2020. Why JavaScript Is the Programming Language of the Future [WWW Document]. *freeCodeCamp.org*. URL <https://www.freecodecamp.org/news/future-of-javascript/> (accessed 8.29.21).
- Netscape Communications Corporation, 1995. NETSCAPE AND SUN ANNOUNCE JAVASCRIPT, THE OPEN, CROSS-PLATFORM OBJECT SCRIPTING LANGUAGE FOR ENTERPRISE NETWORKS AND THE INTERNET. MOUNTAIN VIEW, California.
- Nuntasen, N., Innet, S., 2007. A Novel Approach of Genetic Algorithm for Solving University Timetabling Problems: a case study of Thai Universities 7.
- The Linux Foundation, 2018. 2018 NODE.JS USER STUDY DETAILED REPORT OF FINDINGS.
- Tilkov, S., Vinoski, S., 2010. Node.js: Using JavaScript to Build High-Performance Network Programs. *IEEE Internet Comput.* 14, 80–83. <https://doi.org/10.1109/MIC.2010.145>
- Tsoukalas, L., Uhrig, R., Zadeh, L., 1997. *Fuzzy And Neural Approaches in Engineering* | Wiley. Wiley.
- Wittern, E., Ying, A.T.T., Zheng, Y., Laredo, J.A., Dolby, J., Young, C.C., Slominski, A.A., 2017. Opportunities in Software Engineering Research for Web API Consumption, in: 2017 IEEE/ACM 1st International Workshop on API Usage and Evolution (WAPI). Presented at the 2017 IEEE/ACM 1st International Workshop on API Usage and Evolution (WAPI), IEEE, Buenos Aires, Argentina, pp. 7–10. <https://doi.org/10.1109/WAPI.2017.1>
- Yavirac, n.d. URL <http://yavirac.edu.ec/> (accessed 8.26.21).