

A Meta-Heuristic Based on Genetic Algorithm for Selecting Bailiffs by Districts

Ribamar Louira do Carmo¹, Omar Andres Carmona Cortes²,
Fernando Jorge Cutrim Demétrio¹

¹ Universidade Estadual do Maranhão, Computer Engineer Department,
São Luis, MA, Brazil

² Instituto Federal do Maranhão, Informatics Department,
São Luis, MA, Brazil

ctrinfo.ribamar@gmail.com, fdemetrio@uema.br, omar@ifma.edu.br

Abstract. This paper presents a proposal for solving the Problem of Selecting Bailiffs by Districts (PSB/D) in the central of warrants in Maranhão-Brazil using Genetic Algorithm (GA). The complete solution to the PSB/D problem is a Web-based software called GAPSB/D. We conducted several experiments to prove its applicability considering two scenarios. The first scenario represents the central of warrants in the city of Imperatriz. The second one depicts the city of São Luís. The GA results are compared against two approaches: the manual selection currently in use and Simulated Annealing (SA). Results have proved GA could create solutions as good as the manual process, and it is more stable than SA, especially in the second scenario, in which the search space can reach a size of $\approx 10^{71}$ possibilities of solutions.

Keywords: Bailiff, np-hard, metaheuristic, genetic algorithm, simulated annealing.

1 Introduction

Real world problems are usually complex and, in most cases, enumerative algorithms are impossible to be applied due to constraints and the size of search space. A real-world problem we have faced is the selection of bailiffs by Districts (PSB/D), which is ruled by the Provision 18/2011 of General Internal Affairs of Justice of State of Maranhão (CGJ-MA in Portuguese) [7]. The 2nd Art. of that provision says the allocation of bailiffs in their respective areas (districts) will be done quarterly and electronically by the information technology stall of the Court of Justice.

Currently, the selection and distribution is done manually by a Microsoft Excel spreadsheet, and all constraints are also met manually in a trial-error based; this process takes about two days and, at least, two employees to perform the four distributions of 1 year. Also, the search space is too large, about 10^{22}

possibilities of solutions for the first scenario (Imperatriz) and 10^{71} for the second one (São Luis). In fact, that problem size becomes attractive the use of meta-heuristics such as Genetic Algorithms.

In this context, this paper proposes an optimization model for the PSB/D problem. Thus, it can be solved by any binary meta-heuristic. At the best of our knowledge, there is no other way to solve this problem in the literature neither using a meta-heuristic nor a simplified enumerative algorithm. In fact, the works [2,5,8,12,16,18,19,20] solve problems in the legal area, but they are entirely different problems. In other words, there are no applications related to our problem. Thereby, our proposal is to solve the problem using a Genetic Algorithm (GA) [4,13,14]. Indeed, we used a specific GA called selection-mutation GA [1]. The final application is a Web-based one called GAPSB/D, which was developed using a combination of open source tools, techniques, models, and patterns. In fact, the solution will be available via the Web for using in different cities including those one mentioned in both test scenarios. Furthermore, we have compared the GAPSB/D with both the manual process and Simulated Annealing. Results have shown GAPSB/D produces reliable solutions with high quality.

The remainder of this paper is divided as follows: Section 2 presents some important concepts about the distribution problem. Section 3 details our proposal presenting how individuals are modeled and how to compute the fitness function; Section 4 shows all experiments; finally, Section 5 presents the conclusions and future work.

2 Distributing Bailiffs

Some concepts are necessary to understand how bailiffs are distributed. The first one is the concept of a district, which is composed by one or more neighborhoods. A Central is a place where writs and court orders are managed. A bailiff is an officer who belongs to a district and is responsible for delivering writs and court orders. Figure 1 shows the relationship between these concepts.

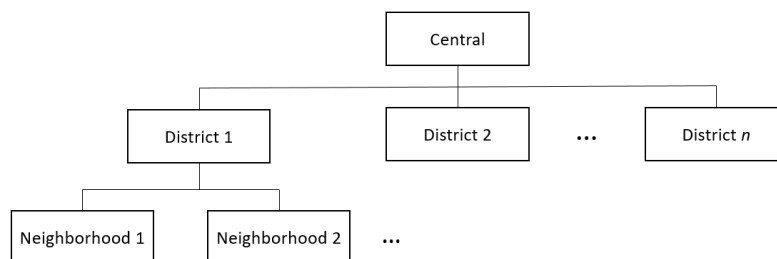


Fig. 1. Relationship between concepts.

In this work, we deal with two different scenarios. The first one is the Central of Imperatriz, which is devised by 27 bailiffs and 6 districts. The second scenario is the Central of São Luís that is composed of 16 districts and 84 bailiffs. Thus, the problem is how to distribute all bailiffs in order to deliver all writs and court orders.

This process is round-robin based and done quarterly; however, some rules must be obeyed. In the first one, none of the bailiffs can be allocated in the same district sequentially, *i.e.*, a bailiff can not be assigned to the same district in contiguous quarters. Further, there is a constraint that imposes that a district has an exact number of bailiffs. The distribution must select exactly the amount of available bailiffs on each district; in other words, none of the bailiffs can be idle.

3 Our Proposal: GAPSB/D

The Genetic Algorithm for Selecting Bailiffs by Districts (GAPSB/D) is a selection-mutation genetic algorithm for solving the PSB/D. The main characteristic of this approach is the unusual method of creating the initial population, which uses a constructive algorithm to try to select the best individuals. Similar operators can be seen in the works [9] and [6]. Furthermore, the GAPSB/D adds two different fitness functions: the population fitness and the genes fitness. Both are set using a minimum value, and both are used as stop criteria.

Because the GAPSB/D is a selection-mutation GA, the crossover operator is not used. Moreover, tests indicated the crossover operator created many invalid individuals that had to be submitted to repairing method in this particular application, increasing the computational time considerably. In other words, the benefit of the crossover operator does not pay-off the required processing. Algorithm 1 shows the pseudocode of the GAPSB/D.

Algorithm 1: The Genetic Algorithm GAPSB/D.

```
1 begin
2   Create initial population
3   Repair invalid individuals
4   Evaluate initial population
5   while stop criteria not achieved do
6     Run genetic mutation
7     Repair invalid individuals
8     Evaluate new population
9     Update population
10  end
11 end
12 return Best Population
```

The initial population is created using a constructive algorithm [9] that keeps individuals feasible by repairing the invalid ones. The stop criterion obeys one of the following conditions: (i) when the algorithm reaches the minimum of the population fitness, and all genes of the population met the gene fitness; or, (ii) when the algorithm reaches the maximum number of iterations.

While the stop criterion is not reached, the population undergoes mutation. Any invalid individual goes to the repairing function, which runs for 100 iterations trying to repair it. If the repairing process fails, the individual is added to the population nevertheless. Then, the population is evaluated, and the updating process only occurs if the new population (population fitness) is better than the previous one. Next sections give details about the representation and explain each step of the GAPSB/D algorithm.

3.1 Individual Representation

The chromosome representation is fundamental to the AG. It is a way of transforming the information of the problem into a solution that a computer can deal with. In this context, each chromosome or individual is binary and has embedded into itself the following information:

- An individual represents a district;
- A gene represents a bailiff;
- The number of genes is the number of bailiffs involved in the distribution;
- A gene equals one indicates that the bailiff was selected to that particular district;
- The population represents the complete distribution of the available bailiffs throughout the districts;
- The population size is the number of districts used in the distribution;
- The chromosome size is the number of bailiffs used in the distribution;
- Capacity is the maximum amount of genes equal to 1 that each chromosome must possess. It is a constraint that the algorithm tries to obey because individuals out of the capacity are unfeasible solutions;
- Distribution is the activity of selecting bailiffs and assigning them to the districts.

Table 1 presents an instance of the population in the GAPSB/D. In this particular example, we have six districts, 20 bailiffs, and a population size equals to 6. Moreover, the table shows the gene capacity, which is the maximum number of bailiffs that can be selected by a district. For example, the individual 6 depicts a district with five bailiffs being selected to that district. Furthermore, we can notice that all population is valid in this example.

3.2 Initial Population

The GAPSB/D tries to select only the best genes for each individual when the initial population is created. If the capacity of each chromosome is violated, the

Table 1. Representation of the PSB/D in GA.

Individual	Capacity of genes	Chromosome
Individual 1	3	10000000110000000000
Individual 2	3	00010000000101000000
Individual 3	3	00000000000010110000
Individual 4	3	00001000001000000010
Individual 5	3	00000000000000001101
Individual 6	5	01100111000000000000
Total:	20	

repair method is executed to transform the infeasible solution into a valid one [9]. The process is iterative, and if after 100 iterations of repairing an individual is still invalid, it is added to the population nevertheless.

3.3 Evaluations: Fitness Functions

The fitness function must represent the quality of each individual in a particular problem to be solved [4], [14]. However, as previously mentioned, our approach introduces two more fitness functions: the population and gene fitnesses. As one can imagine, the gene fitness is used to calculate the individual fitness, which is used to compute the population fitness. The gene fitness is calculated using the historical selection of recent districts, *i.e.*; we consider the historical information of which bailiff was selected to which district in chronological order according to Equation 1:

$$ag = \begin{cases} p \times * \frac{100}{ps}, & \text{if } p > 0 \\ 100, & \text{otherwise} \end{cases}, \quad (1)$$

in which p is the position of the district in the history list of the last n districts, n is the number of districts participating in the distribution, and ps is the population size. An $ag = 100$ when $p = 0$ indicates the bailiff has not been selected by the district in the last n distributions. In the end, ag is a vector containing all genes evaluations. Afterward, the individual is evaluated by the lower fitness of its genes. If any chromosome has a gene with zero fitness, then zero will be the individual fitness. Finally the population is evaluated by Equation 2, in which p is the amount of genes with best fitness, *i.e.*, with fitness equals to 100, and cl is the chromosome length:

$$qg = p \times \frac{100}{cl}. \quad (2)$$

3.4 Updating Population and Elitism

Updating the population is a small modification in the population that almost not changes the processing time, but it assures that the performance of the AG always enhances over the generations. A good practice is to choose an individual

randomly combined with an elitist strategy, which ensures that the best solution never perishes while the algorithm tries to avoid the genetic convergence [14].

Replacing the old population by the new population based on the population fitness in GAPS/D does not guarantee the best individual will belong to the new population because the population fitness is a mean fitness. Thus, if the new population is better than the current one, one or two individuals (the best ones) from the current population are selected to go to the next generations. The choice is made randomly between all individuals with fitness equals to 100%. Only one individual is selected in the first scenario and two in the second one.

3.5 Genetic Mutation

The genetic mutation is applied to each individual of the population based on a mutation rate, excepting on those ones that have been selected by the elitism in the previous generation [3], [10]. After several tests, we could notice that a rate of 50% can explore the diversity of individuals and produces quality solutions in both scenarios. On the other hand, this high rate can also produce invalid chromosomes, which are repaired as previously stated.

4 Experiments

The experiments were conducted on a notebook with 4th generation processor, Intel Core i7-4500U, 08 GB of RAM, and 1 TB SATA hard drive (5400 RPM), running the Windows operating System 8.1, 64bits. Furthermore, as previously mentioned, the experiments were performed in two different scenarios using the parameter shown in Table 2 for GA and SA algorithms [17,11], respectively. Ten distributions of Bailiffs were performed for both scenarios. The results represent the mean of the fitness, runtime and standard deviation corresponding to 30 executions.

Table 2. Inputs parameter GAPS/D and SA per scenario.

GA		
Inputs parameter	scenario 1	scenario 2
Mutation rate	50%	50%
Generations number	10.000	30.000
Percentage of best fitness	100.00%	100.00%
Percentage of minimal fitness	100.00%	100.00%
SA		
Inputs parameter	scenario 1	scenario 2
Evaluations number	10.000	30.000
Percentage of best fitness	100.00%	100.00%
Percentage of minimal fitness	100.00%	100.00%

Regarding the comparison against SA, a solution is formed by a chromosome with 120 genes for the first scenario and 1344 genes for the second one. This modification is necessary for two reasons. The first one is for making the comparison fair in terms of evaluations. The second one is for having a full solution, which involves all bailiffs and districts.

In order to claim if the GAPSB/D is better than SA, we performed a bi-caudal t-test with $\alpha = 0.95$ considering H_0 as being “all means are the same”; therefore, any value without the interval $[-2.0452, 2.0452]$ is considered as a rejection of H_0 . A t-test is possible because of the central limit theorem, which states that any sample greater than 30 instances tends to present a normal distribution [15].

4.1 Scenario 1: 6 Individuals x 20 Genes

Table 3 illustrates the problem size of the Central of Warrants of Imperatriz, which has 6 districts and 20 bailiffs. Thus, combining all possibilities of the problem size we obtain a search space of something around 2.9×10^{22} .

Table 3. Scenario 1: illustrates the problem of the Central of Warrants of Imperatriz.

Individual	Genes capacity	Combinations
Individual 1	3	1.140
Individual 2	3	1.140
Individual 3	3	1.140
Individual 4	3	1.140
Individual 5	3	1.140
Individual 6	5	15.504
Total:	20	$\approx 10^{22}$

Figure 2 shows the comparison of the average fitness of 10 distributions for scenario 1, proving that the GAPSB/D presents, in general, better solutions than SA because SA obtained the best fitness only up to the second distribution. It is important to notice as we increase the number of distributions, we also increment the difficulty on solving the problem because it affects the historical of each bailiff.

Table 4 shows a t-test comparing GAPSB/D versus SA. As we can see, after the second distribution we start rejecting H_0 , which means that difference between algorithms starts being meaningful, *i.e.*, GAPSB/D starts presenting better solutions from the third distribution to the last one.

4.2 Scenario 2: 16 Individuals x 84 Genes

The second scenario represents the problem of the distribution of bailiffs in the central of warrants of São Luís. The information presented in Table 5 were extracted from the database of the System of process accompanying, called Themis

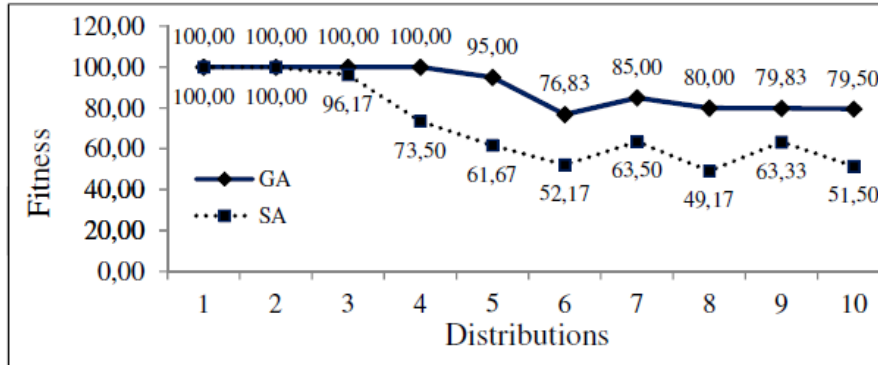


Fig. 2. Scenario 1: Average Fitness.

Table 4. Scenario 1: GAPS/D X SA.

Distributions	Average Fitness			Standard deviation	
	GAPS/D	SA	T-Test	GAPS/D	SA
1	100.00	100.00	0.00	0.00	0.00
2	100.00	100.00	0.00	0.00	0.00
3	100.00	96.17	6.71	0.00	3.13
4	100.00	73.50	29.39	0.00	4.94
5	95.00	61.67	30.84	0.00	5.92
6	76.83	52.17	23.05	4.25	3.39
7	85.00	63.50	29.65	0.00	3.97
8	80.00	49.17	37.00	0.00	4.56
9	79.83	63.33	25.74	0.91	3.56
10	79.50	51.50	35.87	2.01	3.75

(Internal system-use, developed by the IT of Court of Justice). Hence, combining all possibilities of the problem size we obtain a search space of something around $\approx 10^{71}$.

Figure 3 shows the average fitness for 10 distributions, in which we can observe that GAPS/D produces solutions with high quality if compared with SA. After the fourth distribution, the quality of solutions from SA decreases considerably, reaching an average fitness as small as 13.57.

Table 6 presents a comparison for the second scenario, in which we can see the differences start being meaningful from the fourth distribution on. An interesting fact is as the problem becomes harder to solve the GAPS/S remains more stable presenting the most significant differences between the fifth and eighth distribution. The differences on the t-test show the instability of the SA algorithm to cover the search space.

Table 5. Scenario 2: districts of the central of warrants of São Luís.

Individual	Genes capacity	Combinations
Individual 1	3	95.284
Individual 2	5	30.872.016
Individual 3	5	30.872.016
Individual 4	7	4.529.365.776
Individual 5	7	4.529.365.776
Individual 6	5	30.872.016
Individual 7	5	30.872.016
Individual 8	6	406.481.544
Individual 9	3	95.284
Individual 10	10	2.761.025.887.620
Individual 11	8	30.872.016
Individual 12	4	43.595.145.594
Individual 13	3	1.929.501
Individual 14	6	406.481.544
Individual 15	4	1.929.501
Individual 16	3	84
Total:	84	$\approx 10^{71}$

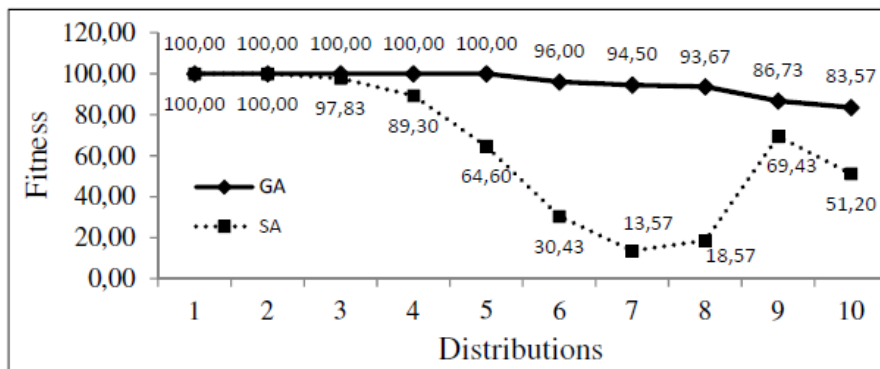


Fig. 3. Scenario 2: average fitness.

5 Conclusion

This work presented a selection-mutation GA for solving the problem of bailiffs distribution by districts. Results have proven that the implemented Genetic Algorithm is stable and reliable because solutions obey all constraint required by the Provision 18/2011. Furthermore, GAPS/D showed better results than Simulated Annealing and reduced considerably the time needed by doing the distribution manually. Moreover, the results indicate the possibility of using GA in bigger instances containing more districts and more bailiffs.

Future work includes testing integer representation providing more flexibility in selecting bailiffs; parallelizing bigger instance of scenarios using more districts

Table 6. Scenario 2: GAPSB/D X SA.

Distributions	Average Fitness			Standard deviation	
	GAPSB/D	SA	T-Test	GAPSB/D	SA
1	100.00	100.00	0.00	0.00	0.00
2	100.00	100.00	0.00	0.00	0.00
3	100.00	97.83	13.57	0.00	0.87
4	100.00	89.30	24.77	0.00	2.37
5	100.00	64.60	6.52	0.00	29.72
6	96.00	30.43	10.16	1.62	35.79
7	94.50	13.57	15.98	1.28	27.76
8	93.67	18.57	12.83	2.07	31.54
9	86.73	69.43	35.02	2.27	2.60
10	83.57	51.20	26.03	3.09	6.40

and bailiffs; test different meta-heuristics such as Differential Evolution and Particle Swarm Optimization.

Acknowledgment. The authors would like to thank FAPEMA (Fundação de Amparo à Pesquisa e ao Desenvolvimento Científico e Tecnológico do Maranhão) for funding this research.

References

1. Berard, J.: Genetic algorithms in random environments: Two examples. *Probability Theory and Related Fields* 133, 123–140 (2005)
2. Bhawna, K., Kumar, G., Bhatia, P.K.: Software test case reduction using genetic algorithm: A modified approach. *international Journal of innovative Science, Engineering & Technology* 3(5), 6 (May 2016)
3. Bolton, C.C., Gatica, G., Barra, C.R., Parada, V.: A multi-operator genetic algorithm for the generalized minimum spanning tree problem. *Expert Systems With Applications* 50, 18 (2016)
4. Cortes, O.A.C., Barros, B.A.S., Lopes, R.F., Silva, J.C.D.: Um algoritmo híbrido paralelo cooperativo baseado em de, pso e ag: Uma avaliação em computadores multicore. *Congresso Brasileiro Em inteligência Computacional (CBIC)* (2013)
5. Cortes, O.A.C., Silva, J.C.: A local search algorithm based on clonal selection. In: *Eleventh Brazilian Symposium On Neural Networks*. p. 6 (2010)
6. Dombry, C.: A weighted random walk model, with application to a genetic algorithm p. 21 (2007)
7. General internal Affairs of Justice, SÃO LuÍS-Ma: Provision 18/2011 (Ago 2011)
8. Gomes, P.V., Saraiva, J.T.: Hybrid genetic algorithm for multi-objective transmission expansion planning p. 6 (2016)
9. Gomes, W.D.P., Gualda, N.D.F.: Modelagem integrada do problema de programação de tripulantes de aeronaves. *Transportes* 19(1), 2332 (Maio 2011)
10. Hassanat, A., Alkafaween, E., Al-Nawaiseh, N.A., Abbadi, M.A., Alkasassbeh, M., Alhasanat, M.B.: Enhancing genetic algorithms using multi-mutations p. 17 (2016)

11. Hu, L., Liu, J., Liang, C., Ni, F., Chen, H.: A phoenix++ based new genetic algorithm involving mechanism of simulated annealing. *international Journal of Distributed Sensor Networks* 2015, 8 (2015)
12. Krause, J., Parpinelli, R.S., Lopes, H.S.: Proposta de um algoritmo inspirado em evolução diferencial aplicado ao problema multidimensional da mochila p. 11 (2012)
13. Linden, R.: *Algoritmos Genéticos*. Brasport, Rio De Janeiro, 2a edn. (2008)
14. Linden, R.: *Algoritmos Genéticos*. Editora Ciência Moderna Ltda, Rio De Janeiro, 3a edn. (2012)
15. Montgomery, D.C., Runger, G.C.: *Applied statistics and probability for engineers* (2003)
16. Singh, V., Sharma, S.K., Vaibhav, S.: Transport aircraft conceptual design optimization using real coded genetic algorithm. *international Journal of Aerospace Engineering* 2016, 11 (2016)
17. Sousa, J., Cunha, M.D.C., Marques, A.S.: A simulated annealing algorithm for the optimal operation of water distribution networks. *Joint international Conference On Computing and Decision Making in Civil and Building Engineering* pp. 14–16 (June 2006)
18. Wang, J., Acharya, S., Kam, M.: Adaptive decision fusion using genetic algorithm p. 6 (2016)
19. Werner, J.C., Fogarty, T.C.: Genetic algorithm applied in clustering datasets. *Scism, South Bank University* p. 6 (2015)
20. Wijayaningrum, V.N., Mahmudy, W.F.: Optimization of ships route scheduling using genetic algorithm. *indonesian Journal of Electrical Engineering and Computer Science* 2(1), 180–186 (April 2016)