Artificial Immune System for Solving Santa's Workshop Tour 2019

Denis Mitana and Miroslav Sumega

Slovak University of Technology in Bratislava, Ilkovičova 2, 842 16 Bratislava, Slovakia

Abstract. Goal of this work is to minimize the penalty cost for the Santa's Workshop Tour 2019 scheduling optimization problem. Scheduling tasks are often NP-hard problems so it is not possible to solve them in a deterministic way and non-deterministic optimization algorithms are needed. For solving this kind of problems nature-inspired algorithms are often used. The proposed nature-inspired optimization method of Artificial Immune System algorithm (AIS) is used for finding optimal day to visit the workshop for each family while minimizing Santa's costs. The AIS algorithm is tested with three and two different methods of mutation and selection, respectively, showing that more informed mutations and less selected Antibodies for the next generation can significantly improve results. Experiments also showed that negative selection is more appropriate than positive selection, which got stuck in the local minimum. Our best achieved score is 2,789,989. The results show that AIS algorithm is suitable for the given scheduling optimization task.

Keywords: Artificial Immune System · Scheduling · Optimization.

1 Introduction

Scheduling is a problem where are many tasks but only a limited time to perform them. Therefore optimization of scheduling aims to assign tasks to specific times in a way that minimizes a fitness function, which might be the cost of performing the given scheduling or time to perform it.

Optimization of scheduling is an NP-hard problem, which is most manifested when there is a large number of tasks. Nature-inspired algorithms are often used to solve such types of problems. They have ability to find a good enough solution in reasonable amount of time. Their another advantage is that they are not problem dependent, therefore one algorithm can be used for solving two entirely different problems.

In this paper we focus on using AIS, which is a nature-inspired algorithm based on the human immune system, to solve the Santa's Workshop Tour 2019 problem. It is a large scheduling problem with various constrains. In section 2 we describe existing solutions to scheduling problems. Section 3 contains information about our implementation of AIS. In section 4 we describe problem that we are solving in detail. Section 5 contains experiments we performed to validate our solution and section 6 summarizes this paper.

2 Related Work

There are various approaches to solving scheduling problems. In [4] they used AIS with 6 different types of mutations to solve the student assignment problem. They evaluated proposed solution on 4 different benchmark student assignment problems and also compared with Genetic Algorithm (GA). Their approach was superior to GA. Makespan in the job shop scheduling was optimized in [2] using AIS. To evaluate their solution they used 130 benchmark problems and compared with Tabu Search Shifting Bottleneck (TSSB). Their solution using AIS achieved best results. Bat Algorithm (BA) parallelized on multiple processors using schemes of communication strategies and makespan was used to solve the job shop scheduling problem [3]. This solution is based on traditional BA, where populations are divided into groups. They used 43 benchmark instances of job shop scheduling with various sizes to compare to traditional BA and Particle Swarm Optimization (PSO). They achieved better results. In [5] they used Beer froth Artificial Bee Colony (BeFABC) algorithm to solve the job shop scheduling problem. Proposed solution was evaluated on 25 benchmark tests and compared to other state-of-the-art algorithms. They also used it to solve 62 well known instances of the job shop scheduling problem. Results showed that the proposed solution was comparable to other algorithms.

3 Artificial Immune System

There are various approaches using nature-inspired algorithms used for solving scheduling problem in literature. We chose to use AIS because it was used to solve scheduling problems and achieved better results than other nature-inspired algorithms.

AIS [1] is a nature-inspired algorithm based on mimicking human immune system. It consists of an Antigen which is considered as an external object invading body and an Antibody which reacts with antigen to suppress it. In other words, the Antibody represents solution to the given task.

3.1 Implementation

We decided to use own implementation of AIS (lst. 3.1), due to more flexibility when experimenting with various settings. In our implementation, we did not consider Antigen, since we solve one specific problem. We consider only the Antibody, which represents all families, each assigned to a specific day.

At lines 1-2 we generate initial population of Antibodies, which will be later changed in body of the loop. And we also calculate their affinity to be able to select new population.

At line 4 we calculate fitness of a population (eq. 1). Then at line 5 we copy each Antibody according to:

$$n\_clones = log_2(fitness_{diff})$$

Listing 3.1: Pseudo-code of our implementation of AIS.

```
 1  Generate initial population of Antibodies
 2  Calculate affinity of population
 3  For each generation do:
 4      Calculate fitness of population
 5      Clone population
 6      Mutate clones
 7      Calculate fitness of clones
 8      Select temporary population
 9      Calculate affinity of temporary population
10      Select new population from temporary population
11      If size of new population < size of population:
12          Generate additional Antibodies
```

where *n_clones* is how many clones of Antibody will be created and $fitness_{diff}$ is a difference between the largest fitness in the population and fitness of Antibody to be cloned. Therefore better solutions have more clones than worse solutions.

Following at line 6 we mutate clones. Mutation is considered as moving a family from one day to another day. We use three different approaches to mutation:

Basic - Random family is moved to random day without considering preferences.

Preference - Random family is moved to a day randomly chosen from its preferences. Order of preferences is not considered. If chosen day is full, a new family is chosen.

Advanced Preference - Random family is moved to a most preferred day, on which there is still place for entire family. Therefore if the most preferred day is full, the second most preferred day is checked, etc. If no day from preferences is available, a new family is chosen.

All these approaches perform multiple mutations on a single Antibody. Number of mutations is obtained by:

$$n\_mutations = \sqrt[3]{fitness}$$

where *n_mutations* is how many mutations of the Antibody will happen and $fitness$ represents a fitness value of the mutated Antibody. Therefore better solutions are mutated less and worse solutions are mutated more.

At line 7 we calculate fitness of clones in order to select the temporary population at line 8. In this selection, for each Antibody, fitness of its best clone is compared to fitness of Antibody. If a clone, which is mutated, is better than the original Antibody, it replaces Antibody, else all clones are thrown away and original Antibody is used. This allows mutations to appear in the population and improve it.

Affinity of the temporary population is calculated at line 9. Then at line 10 a new population, which will be used in next iteration of the loop, is selected. We use two types of selectors:

Basic - Only Antibodies, whose affinity is smaller (negative selection) / larger (positive selection) than specified affinity threshold are selected to the new population.

Percentile affinity - Only Antibodies, whose affinity is smaller (negative selection) / larger (positive selection) than specified percentile of affinities of the entire population are selected to the new population.

Lines 11 and 12 are used for generating additional Antibodies if some were filtered at line 10. This generation is done in order to have population of constant size each iteration.

To perform correct cloning, mutations and selections appropriate values for affinity and fitness values have to be computed. Affinity value between two Antibodies is the number of families that have assigned the same day in both Antibodies [4]. To obtain total affinity value of each Antibody, this operation is executed between each pair of Antibodies. We selected this affinity value in order to prevent keeping of similar Antibodies in the population and thus boost exploration of the optimization. As a fitness value we used *score* (eq. 1) defined by the task. This value defines quality of a solution quite well.

4 Santa's Workshop Tour 2019

Santa's Workshop Tour 2019 is a Kaggle competition[1] that defines scheduling optimization problem. The goal of this task is to schedule the families to Santa's Workshop in a way that minimizes the penalty cost to Santa. Every family must be scheduled for one and only one day. Provided data consists of 5,000 families that have listed their top 10 preferences for the dates they'd like to attend Santa's workshop tour. Dates are integer values representing the days before Christmas. Each family also has a number of people attending

Solution is scored according to the penalty cost to Santa for suboptimal scheduling. The constraints and penalties are as follows:

- the total number of people attending the workshop each day must be between 125 - 300,
- *preference cost* penalty is computed according to families' assigned days relative to their preferences[2],
- *accounting penalty* is cost arisen from many different effects such as reduced shopping in the Gift Shop when it gets too crowded, extra cleaning costs, a very complicated North Pole tax code, etc.

[1] `https://www.kaggle.com/c/santa-workshop-tour-2019/overview`

[2] For detailed description of cost for each choice see `https://www.kaggle.com/c/santa-workshop-tour-2019/overview/evaluation`.

Table 1: Experiments with various parameters of AIS for negative selection. In all of these experiments we used Basic clonator and Percentile selector. Columns *Affinity*, *Generations* represent affinity threshold of the selector and number of generations, respectively.

No.	Mutator	Affinity	Population size	Generations	Min fitness
1	Preference	50	10	10	7,763,539
2	Preference	50	20	10	7,803,577
3	Preference	25	10	10	7,962,905
4	Preference	75	10	10	7,864,233
5	Preference	50	10	100	7,818,492
6	Advanced Pref.	50	10	10	7 299 069
7	Advanced Pref.	50	10	100	**3 861 740**

Mentioned *accounting penalty* is defined as:

$$\sum_{d=100}^{1} \frac{N_d - 125}{400} * N_d^{\left(\frac{1}{2}+\frac{N_d - N_{d+1}}{50}\right)}$$

where N_d is the occupancy of the current day, and N_{d+1} is the occupancy of the previous day[3]. Initial conditions are $d = 100$ and $N_{101} = N_{100}$, since it starts on the date 100 days before Christmas and ends on Christmas Eve. Final evaluation score is computed as:

$$score = preference\ cost + accounting\ penalty \tag{1}$$

5 Experiments

In order to evaluate how good is our solution of Santa's Workshop Tour 2019, we performed multiple experiments, each with different parameters of AIS algorithm. We compared these experiments by comparing fitness of the best solution.

5.1 Experiments with Negative Selection

In first 2 experiments (table 1) we can see that population size does not have any significant impact on quality of found solution.

In experiments 3 and 4 we tried to influence the ability of AIS to explore solution space. We expected that affinity threshold of 25th percentile will boost exploration of AIS since only most different solutions are selected, however no improvement was achieved. Compared to that affinity threshold of 75th percentile should have less influence on exploration, but that also did not improve results.

Because we achieved best results so far with affinity threshold of 50 %, we tried bigger experiment (table 1 No. 5, fig. 1). However, results did not improve,

[3] Counting is backward from Christmas.

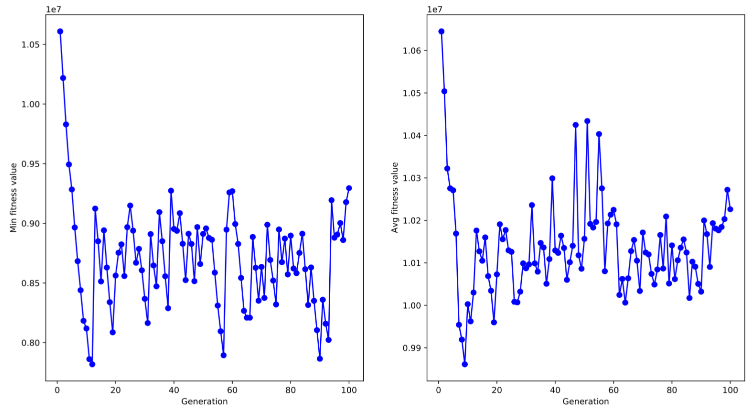

Fig. 1: Experiment No. 5 from table 1. We can see that quality of best solution fluctuates because of weak ability to exploit via mutations.

Table 2: Experiments with various parameters of AIS for positive selection. In all of these experiments we used Basic clonator and Percentile selector. Columns *Affinity*, *Generations* represent affinity threshold of the selector and number of generations, respectively.

No.	Mutator	Affinity	Population size	Generations	Min fitness
1	Preference	50	10	10	9,846,231
2	Preference	25	10	10	**9,438,700**
3	Preference	75	10	10	10,170,092
4	Advanced Pref.	50	10	10	9,721,301
5	Advanced Pref.	50	10	100	9,620,380
6	Advanced	50	10	100	9,762,966

because mutations were not performed effectively and therefore it was very hard to improve best solution. To surpass this problem, we implemented *Advanced Preference Mutator*. With this mutator, we achieved best results so far in 10 generations (table 1 No. 6), therefore we tried to use it on 100 generations (table 1 No. 7, fig. 2). We can see, that this kind of mutator greatly improved ability of AIS to find good solutions.

5.2 Experiments with Positive Selection

Experiments with positive selections were targeted to improve exploitation of AIS algorithm. Since positive selection selects only similar Antibodies to next generations, it searches a smaller portion of a solution space and thus tries to utilize it better. However, this approach was not successful (table. 2). Even if *Advanced Preference* mutator outperformed *Preference* mutator, similar to negative selection experiments, it is only a minor improvement. Also more generations were not useful. We did not use positive selection in other experiments.

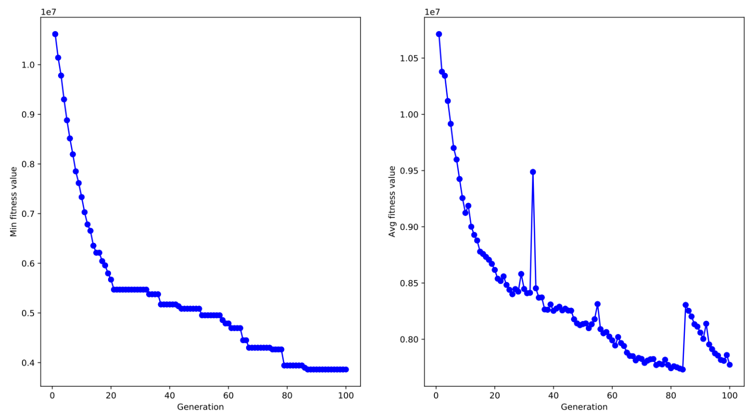

Fig. 2: Experiment No. 7 from table 1. Quality of best solution steadily improves because of improved mutations.

Table 3: Large experiments with different affinity threshold settings for negative selection. In all of these experiments we used Basic clonator and Percentile selector. Columns *Affinity*, *Generations* represent affinity threshold of the selector and number of generations, respectively.

No.	Mutator	Affinity	Population size	Generations	Min fitness
1	Advanced Pref.	50	20	200	6,770,944
2	Advanced Pref.	25	20	200	**2,789,989**

5.3 Large Experiments

At last we performed large experiments with best parameters settings obtained from previous experiments (table 3). Maximal used population size was only 20, because computational complexity of our solution is $\mathcal{O}(n^2m)$, where n is the population size and m is the number of generations and we were limited by time and resources. In this settings experiment ran approx. 8 hours. When using 50th percentile as *Affinity threshold* (table 3 No. 1), which achieved best result in first experiment, optimization started to oscillate in 17th generation and it did not improve anymore (fig. 3). We assumed that it happened due to large *Affinity threshold*, which allowed half of the population to be passed to the next generation, but we are not sure about it. Therefore we tried 25th percentile as *Affinity threshold* (table 3 No. 2) and it was successful. Throughout this optimization there were no oscillations and fitness was non-increasing. (fig. 4) and we suppose that it would continue.

5.4 Evaluation

Achieved results proven that AIS is usable for Santa's Workshop Tour optimization task. Experiments shown that more informed mutations can be very helpful.

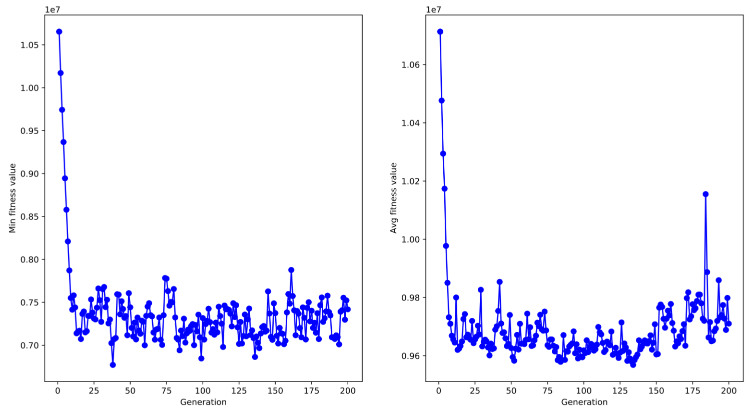

Fig. 3: Experiment No. 1 from table 3. Large affinity decreased ability to select potentially good solutions, therefore best solution fluctuates and does not improve.

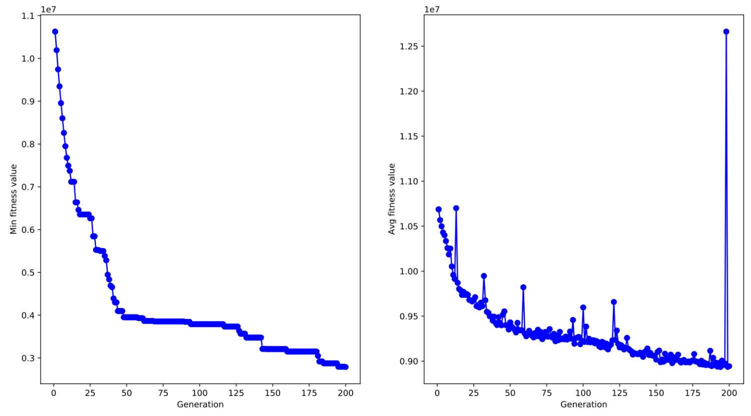

Fig. 4: Experiment No. 2 from table 3. Smaller affinity improved ability to select potentially good solutions, therefore best solution improved steadily.

Also passing less Antibodies to the next generation helps to improve results. On the other hand, the higher utilization of the exploitation using positive selection resulted in the AIS algorithm probably getting stuck in the local optimum.

In comparison with Kaggle's leaderboard[4] our solution is far from the best. Best score at Kaggle is 68,888.04 and there is a lot of submissions achieving this score, so it looks like some very good local or even global optimum. Although our best score, which is 2,789,989, is a lot worse, we were not able to perform really large long-running experiment due to time and computational limitations.

[4] `https://www.kaggle.com/c/santa-workshop-tour-2019/leaderboard`

However, we assume using larger population and more generations would result in better performance.

6 Conclusion

In this work, we presented solution for Santa's Workshop Tour 2019 scheduling optimization problem based on AIS optimization algorithm. We experimented with three and two different approaches to mutation and selection, respectively. Although we achieved a much worse score than the best solutions at Kaggle, we assume that our AIS based solution is correct and it would achieve competitive results in a long-running experiment. Some of the challenges left for the future are to try different functions for obtaining number of clones and modify selection approach for greater use of exploitation with growing generation.

References

1. Aickelin, U., Dasgupta, D.: Artificial immune systems. In: Search methodologies, pp. 375–399. Springer (2005)
2. Chandrasekaran, M., Asokan, P., Kumanan, S., Balamurugan, T., Nickolas, S.: Solving job shop scheduling problems using artificial immune system. The International Journal of Advanced Manufacturing Technology **31**(5-6), 580–593 (2006)
3. Dao, T.K., Pan, T.S., Pan, J.S., et al.: Parallel bat algorithm for optimizing makespan in job shop scheduling problems. Journal of Intelligent Manufacturing **29**(2), 451–462 (2018)
4. El-Sherbiny, M.M., Ibrahim, Y.M.: An artificial immune algorithm with alternative mutation methods: applied to the student project assignment problem. In: International conference on innovation and information management (ICIIM2012), Chengdu, China (2012)
5. Sharma, N., Sharma, H., Sharma, A.: Beer froth artificial bee colony algorithm for job-shop scheduling problem. Applied Soft Computing **68**, 507–524 (2018)