International Journal of Pure and Applied Mathematics
Volume 115 No. 4 2017, 801-812
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
doi: 10.12732/ijpam.v115i4.13

ijpam.eu

GENETIC ALGORITHM ADOPTING IMMIGRATION OPERATOR TO SOLVE THE ASYMMETRIC TRAVELING SALESMAN PROBLEM

Chakir Tajani[1 §], Otman Abdoun[2], Ahmed Idrissi Lahjouji[3]

[1,2]Department of Mathematics
Polydisciplinary Faculty of Larache
Abdelmalek Essaadi University, MOROCCO

[3]Department of Mathematics and Informatics
Faculty of Sciences of Meknes
Moulay Ismail University
MOROCCO

Abstract: In this work, we are interested in improving the performance of genetic algorithm (GA) to solve the Asymmetric Traveling Salesman Problem (ATSP). Several approaches have been developed with genetic algorithms based on the adaptation and improvement of different standard genetic operators. We proposes a new GA adopting immigration strategies to maitain diversity and to perform more the genetic algorithm. Experimental results on series of standard instances of ATSP show that the proposed structured memory immigration scheme in GA effectively improves the performance of GAs.

AMS Subject Classification: 80M50, 90C27, 90B06
Key Words: asymmetric traveling salesman problem, genetic algorithm, optimization problem, immigration operator

1. Introduction

The ATSP can be formulated by an Integer Linear Programming (ILP) model

Received: February 18, 2017
Revised: May 8, 2017
Published: August 8, 2017

[§]Correspondence author

[1] utilizing n^2 binary variables x_{ij} as follow:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{1}$$

$$\sum_{i=1}^{n} x_{ij} = 1, j = 1, ...n \tag{2}$$

$$\sum_{j=1}^{n} x_{ij} = 1, i = 1, ...n \tag{3}$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1, S \subset V, S \neq \emptyset \tag{4}$$

$$x_{ij} \in \{0, 1\}, i, j = 1, ..., n, \tag{5}$$

x_{ij} is equal to 1 if and only if arc(i, j) $(i = 1, ..., n; j = 1, ..., n)$ is in the optimal tour.

(2) and (3) impose that the in-degree and out-degree of each vertex, respectively, is equal to one.

(4) are subtour elimination constraints and impose that no partial circuit exists.

Otherwise, in the asymmetric traveling salesman problem, one is given a set of N cities and for each pair of cities c_i, c_j a distance $d(c_i, c_j)$ with $d(c, c^{'}) \neq d(c^{'}, c)$. The goal is to find a permutation of the cities that minimizes:

$$\sum_{i=1}^{N-1} d(c_{\pi(i)}, c_{\pi(i+1)}) + d(c_{\pi(N)}, c_{\pi(1)}) \tag{6}$$

The ATSP problem is classified as an NP-complete problem [2]. Among the meta-heuristics most used to solve this kind of problem, we find the Genetic Algorithms (GAs) [3] which are evolutionary methods developed by Holland [4]. The population of a GAs evolves by using genetic operators inspired by the evolutionary in biology [5]. It is well known that GAs get stuck in local optima very often. One efficient way of avoiding this problem is maintaining the diversification in population. Then, an immigration operator can be applied in addition to the usual genetic operators which consists in randomly generating a finite number of individuals at regular intervals to replace a substantial percentage of the population [6].

Nevertheless, we agree that immigrants bring progress to any population. Moreover, the more different the immigrants are, the more progress and knowledge is brought. That is why we propose an immigration technique in which the immigrants are not random, but we adopt a technique based on structured immigration which consists in benefiting individuals not inserted during the previous generations (resulting from the crossover and mutation operators of the selected individuals). Thus, a percentage of the most powerful individuals will immigrate after an interval of time instead of the same number of the lowest individuals in the last generation. The complexity of immigration is decreased by executing it only every several generations.

In this work, to resolve ATSP by the GA method, we will present each individual of population by the most adapted method of data representation which is the path representation method. A crossover and mutation operator adapted to the ATSP problem are used in addition to structured immigration operator in order to bring a dynamism and then a diversity to the current population to perform the algorithm and obtain a best optimal solution in a reduced number of iterations.

This paper is organized as follows: The standard genetic algorithm to solve the ATSP is presented in section 2. In Section 3, the genetic algorithms with random immigration process and with the developed immigration operator are established. In Section 4, computational experiment were performed through many ATSPs standard instances. The comparison with the results obtained with standard GA and with GA adopting randam immigration shows that introducing immigration operator to GA for ATSP improves the performance of a GA and provides better solutions in less iterations.

2. Standard genetic Algorithm

The GA is a one of the family of evolutionary algorithms which attracted the interest of many researchers, starting with Holland, who developed the basic principles of genetic algorithm, and Goldberg has used these principles to solve a specific optimization problems. Other researchers have followed this path [7], [8].

In a genetic algorithm a population of individuals (possible solutions) is randomly selected. These individuals are subject to several operators inspired by the evolutionary in biology, called genetic operators (selection, crossover, mutation and insertion) to produce a new population containing in principle better individual. This population evolves more and more until a stopping criterion is satisfied and declaring obtaining optimal best solution. Thus; the performance of genetic algorithm depends on the choice of operators [9], [10].

Several works are focused on the improvement of genetic operators, which has allowed the development of several adapted crossover operators to ATSP and the comparison of their performance, and even the hybridization between two operators to benefit of their specificity and make the GA more efficient.

In this paper, we introduce some known operators presented below: **a. Representation method:** In this work, we consider the resolution of the ATSP by genetic Algorithms where we will present each individual by the most adapted and natural method of data representation, the path representation method, which is the most natural representation of a tour (a tour is encoded by an array of integers representing the successor and predecessor of each city) [4].

3	5	2	9	7	6	8	4

Figure 1: coding of a tour (35297684).

b. UPMX crossover: The uniform partially matched crossover developed by [11], uses the technique of PMX. Any times, it does not use the crossover points; instead, it uses a probability of correspondence for each iteration. The algorithm is described in Fig. 2.

Input: Parents $x_1=[x_{1,1},x_{1,2},\ldots\ldots,x_{1,n}]$ and $x_2=[x_{2,1},x_{2,2},\ldots\ldots,x_{2,n}]$
Output: Children $y_1=[y_{1,1},y_{1,2},\ldots\ldots,y_{1,n}]$ and $y_2=[y_{2,1},y_{2,2},\ldots\ldots,y_{2,n}]$

Initialize

- $y_1 = x_1$ and $y_2 = x_2$;
- Initialize p_1 and p_2 the position of each index in y_1 and y_2;
- Choose two crossover points a and b such that $1 \leq a \leq b \leq n$;

For each i between 1 and n do
  Chose a random number q between 0 and 1;
  if $q \geq p$ then
    $t_1 = y_{1,i}$ *and* $t_2 = y_{2,i}$;
    $y_{1,i} = t_2$ *and* $y_{1,p1,t1} = t_1$;
    $y_{2,i} = t_1$ *and* $y_{2,p2,t2} = t_2$;
    $p_{1,t1} = p_{1,t2}$ *and* $p_{1,t2} = p_{1,t1}$;
    $p_{2,t1} = p_{2,t2}$ *and* $p_{2,t2} = p_{2,t1}$;
  endif
endfor

Figure 2: Algorithm of UPMX Crossover.

c. TWORS mutation: with two mutation points which allows the exchange of position of two genes randomly chosen [12].

Figure 3: Example of TWORS mutation.

d. Insertion Method: We used the method of inserting elitism that consists in copy the best chromosome from the old to the new population. This is supplemented by the solutions resulting from operations of crossover and mutation, in ensuring that the population size remains fixed from one generation to another.

3. Genetic Algorithm with immigration strategies:

3.1. Standard Genetic Immigration Algorithm - SIG:

In random immigrant scheme, the randomly created individuals are inserted into the population by replacing the worst individuals or some individuals selected randomly. The random immigrant scheme increases the diversity by the immigrants continuously introduced. It maintains the diversity level of the population through substituting some individuals of the current population with random individuals every generation. As to which individuals in the population should be substituted, usually there are two strategies: replacing random individuals or replacing the worst ones [13]. In order to avoid that random immigrants disrupt the ongoing search progress too much, especially during the period when the environment does not change, the ratio r_i of the number of random immigrants to the population size n is usually set to a small value.

The pseudo-code for the standard GA with random immigrants investigated in this paper, denoted SIG, is also shown in Fig. 4, where random immigrants replace worst individuals in the population, p_x is the crossover probability, and p_m is the mutation probability.

3.2. Improved Genetic Immigration Algorithm- AIG:

By inspiration from the flux of immigrants that wander in and out of a population between two generations in nature. The random immigration proposed by [14] maintains the diversity level of the population through replacing some individuals of the current population with random individuals, called random immigrants, every generation. As to which individuals in the population should be replaced. But, in order to benefit of the previous generations and of some

```
begin
Initialize population P randomly with constraints validation
evaluate population P
for(iitr=1; iitr<=iter; iitr++)
Sel:=Select For Reproduction(P)   // N individuals
CX := Crossover(Sel, Px)          // Px is the crossover probability
Mut := Mutate(CX, Pm)             //Pm is the mutation probability
Evaluate new individuals Mut// EvaluteMut and sort in descendant
    P' = Elitisme(Mut(1; N/2))    // Perform elitism
If mod( iitr, ItInser) == 0 then            // Execution of SIG
        Generate n random immigrants
        evaluate these random immigrants
        replace the worst individuals in P
EndIf
EndFor
end
```

Figure 4: Algorithm of Approach SIG.

individuals that not be able to be introduced in the population. After a defined interval of time (some generation), we give chance of the best individual to immigrate to the new population. This new operator is called "structured memory immigration operator" shown in Fig. 5.

This way, the introduced immigrants are more adapted to the current environment than random immigrants. Then, the new operator introduces a diversity of the population and more dynamism and exploration of different probable solutions of the problem, in order to obtain a better optimal solution compared to that obtained by the standard genetic algorithm and for the genetic algorithm based on the immigration operator.

4. Computation experiment

To show the effectiveness of the immigration approach (AIG) proposed in this paper, we used standard instances of the ATSP library [15]. Citing all FTV instances (FTV33, FTV35, FTV38, FTV44, FTV47, FTV55, FTV64, FTV70, FTV170) And even other ATSP instances such as: BR17, FT53, FT70, KRO124P, P43 and RY48P. The performance of the new approach is demonstrated through comparisons performed with a uniform genetic algorithm (UGA) and with a standard immigration (SIG). The approach is developed entirely in C ++ language, and turned on an Intel Corei31.7 GHz machine and

```
begin
Initialize population P randomly with constraints validation;
evaluate population P;
for(iitr=1; iitr<=iter;iitr++)
Sel:=Select For Reproduction (P)     // N individuals
CX := Crossover (Sel, Px)     // Px is the crossover probability
Mut := Mutate (CX, Pm)                //Pm is the mutation probability
Evaluate new individuals Mut// EvaluteMut and sort in descendant
    P' = Elitisme(Mut(1; N/2))          // Perform elitism
ImPop = (Mut(N/2; N))
ifiitr%ItInsert == 0 then               // Execution of SIG
        evaluate immigrants subpopulation ImPop
        replace the n worst individuals in P
EndIf
EndFor
end
```

Figure 5: Algorithm of the proposed Approach AIG.

4GB of RAM. The parameters of different approaches deployed in this paper are presented in the table 1. below:

	Npop ; Sel ;X ; P_x ; M ; P_m ; Insert
Uniform Genetic Algorithm (UGA)	10 ; Roulette ; UPMX ; 0,6 ; Twors ; [0.001, 0.2] ; Elitism
Standard Immigration Genetic (SIG)	10 ; Roulette ; UPMX ; 0,6 ; Twors ; [0.001, 0.2] ; Elitism + Standard Immigration (SIG) after *n* iterations
Improved Immigration Genetic (AIG)	10 ; Roulette ; UX ; 0,6 ; Twors ; [0.001, 0.2] ; Elitism + Improved Immigration (AIG) after *n* iterations

Table 1. The parameters for different approaches.

	UGA		SIG		AIG	
	Init	Bsol	Init	Bsol	Init	Bsol
FTV33	3981	3281	3889	2842	3939	2709
FTV35	4698	3688	4629	3281	4484	3127
FTV38	4400	4148	4760	3523	4839	3412
FTV44	5647	5092	5677	4285	5594	3755
FTV47	6281	5594	6492	4647	6648	4630
FTV55	6956	6225	6815	5127	6775	4659
FTV64	8226	7545	8229	6311	8421	6201
FTV70	9301	8086	9244	7173	9254	6891
FTV170	25078	23180	25260	21598	25767	21241

Table 2. optimal solution after 100 iterations for ATSP series: FTV

Table 2 presents the numerical values retrieved after the implementation of the new approach proposed in this paper and the similar values generated by standard immigration (SIG) and the UGA Uniform algorithm on all FTV-ATSP instances after 100 iterations.

With the application of the three UGA, SIG and AIG methods to the resolution of the FTV-ATSP instance series: FTV33, FTV35, FTV38, FTV55, FTV70 and FTV170 is shown in Fig. 6 (a), Fig. 6 B), Fig. 6 (c), Fig. 6 (d), Fig. 6 (e) and Fig. 6 (f).

The representative curves of Fig. 6 reveal the importance of the effective integration of genetic immigration AIG and its influence on the convergence towards the optimal solution compared to the uniform algorithm UGA and even effective if we compare it with standard immigration SIG, which shows the effectiveness of the immigration approach AIG which is the objective of this paper.

	UGA		SIG		AIG	
	Init	Bsol	Init	Bsol	Init	Bsol
BR17	194	50	201	50	157	44
FT53	25386	22242	24086	20291	25240	18566
FT70	69850	66852	67710	62374	70141	61617
KRO124P	187928	158294	171216	132808	179636	127095
P43	22604	16856	22756	11415	22579	11334
RY48P	51734	39395	50747	33604	51001	32102

Table 3. Optimal solution after 100 iterations for other ATSP series

To test the functioning of the proposed approach to resolve ATSP instances with different sizes, we have varied the deployment of the proposed approach with large instances (Table 3), comparing its generated numerical result with the other approaches UGA and SIG.

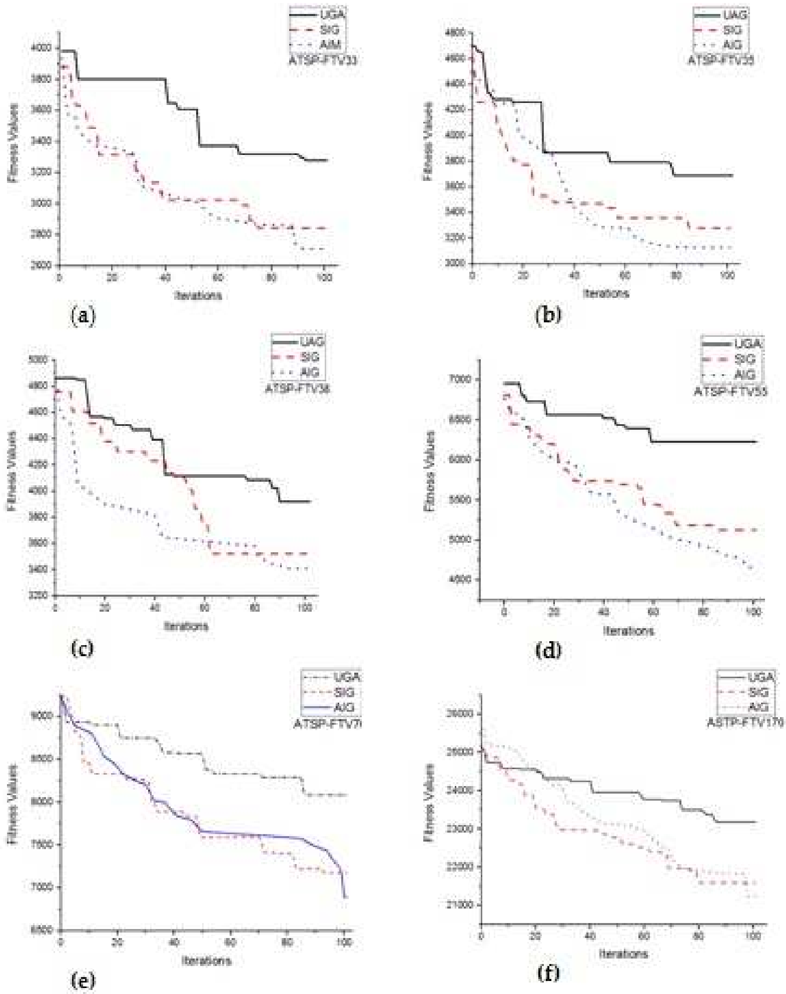

Figure 6: Optimum solution according to number of iterations for ATSP-FTV instances: (a)FTV33 ; (b)FTV35; (c) FTV38; (d) FTV 55; (e) FTV 70; (f) FTV 170.

Fig. 7 emphasize the effectiveness of the proposed AIG approach in solving all variants of the ATSP problem with even large instances, Fig. 7-e and Fig. 7-f, in a reasonable iteration number.

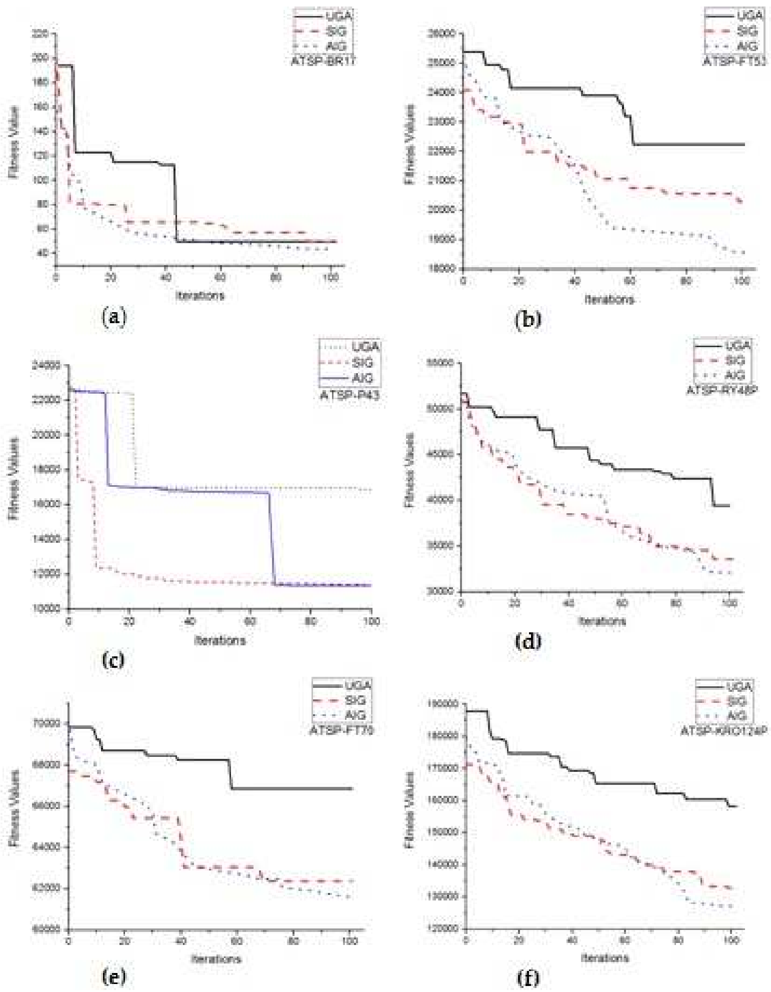

Figure 7: Optimum solution according to number of iterations for other ATSP instances: (a) BR 17; (b) FT 53; (c) P43; (d) RY48P; (e) FT 70; (f) KRO 124P.

5. Conclusion

In this paper, we are interested in performing the genetic algorithm (GA) to solve the ATSP problem which is a NP-complete problem of great importance. Thus, in addition to standard GA operators, an immigration operator based on the insertion of a percentage of best individuals from previous generations, not inserted in previous populations, into the new population after each interval of time. The results obtained for different series of standard instances of ATSP show the effectiveness and robustness of the new proposed immigration procedure to produce dynamism and diversity to the population and provides a better optimal solution in less iterations compared to the standard GA and the GA with random immigration.

References

[1] G. Dantzig, D. Fulkerson and S. Johnson, Solutions of a large-scale traveling-salesman problem, *Journal of the Operations Research Society of America*, **2** (1954), 393-410.

[2] M. R. Garey and D.S. Jonhson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman (1979).

[3] K. Helsgaun, An effective implementation of the Lin-Kernighan traveling salesman heuristic, *Eur. J. of Oper. Res.*, **126** (2000), 106-130.

[4] I. M. Oliver, D. J. Smith and JRC. Holland, A study of permutation crossover operators on the traveling salesman problem, *In Proc. of the second international conference on genetic algorithms (ICGA'87)* Cambridge, MA: Massachusetts Institute of Technology (1987).

[5] D. Goldberg, *Genetic Algorithm in Search, Optimization, and Machine Learning*, Addison Wesley (1989).

[6] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA (1992).

[7] L. Davis, D. Orvosh, A. Cox and Y. Qiu, A Genetic Algorithm for Survivable Network Design, *ICGA* (1993), 408-415.

[8] Z. Michalewicz, *Genetic algorithms + data structures = evolution programs*, Berlin: Springer (1992).

[9] O. Abdoun, C. Tajani and J. Abouchabaka, Analyzing the Performance of Mutation Operators to Solve the Travelling Salesman Problem Analyzing, *Int. J. of Emer. Sci.*, **2** (2012), 61-77.

[10] O. Abdoun, J. Abouchabaka, A Comparative Study of Adaptive Crossover Operators for Genetic Algorithms to Resolve the Traveling Salesman Problem, *Int. J. of Comp. Appl.*, **31** (2011), 49-57.

[11] V. A. Cicirello, Non-wrapping order crossover: An order preserving crossover operator that respects absolute position, *GECCO*, (2006), 1125-1131.

[12] O. Abdoun, C. Tajani and J. Abouchabaka, Hybridizing PSM and RSM Operator for Solving NP-Complete Problems: Application to Traveling Salesman Problem, *Int. J. of Comp. Sci. Iss.*, **9** (2012), 374-378.

[13] L. N. Xing, Y. WuChen, K. WeiYang, F. Hou, X. ShiShen and H. PingCai, A hybrid approach combining an improved genetic algorithm and optimization strategies for the asymmetric traveling salesman problem, *Eng. Appl. of Art. Intel.*, **21** (2008), 1370-1380.

[14] J. Grefenstette, Genetic algorithms for changing environments, *Parallel Problem Solving from Nature II*, (1992), 137-144.

[15] Asymmetric Traveling Salesman Problem Data, http://www2.iwr.uni heidelberg.de/groups/com opt/software/TSPLIB95/atsp/ 2011.