World Academy of Science, Engineering and Technology
International Journal of Computer and Systems Engineering
Vol:17, No:2, 2023

Transfer Knowledge from Multiple Source Problems to a Target Problem in Genetic Algorithm

Tami Alghamdi, Terence Soule

Abstract—To study how to transfer knowledge from multiple source problems to the target problem, we modeled the Transfer Learning (TL) process using Genetic Algorithms as the model solver. TL is the process that aims to transfer learned data from one problem to another problem. The TL process aims to help Machine Learning (ML) algorithms find a solution to the problems. The Genetic Algorithms (GA) give researchers access to information that we have about how the old problem is solved. In this paper, we have five different source problems, and we transfer the knowledge to the target problem. We studied different scenarios of the target problem. The results showed that combined knowledge from multiple source problems improves the GA performance. Also, the process of combining knowledge from several problems results in promoting diversity of the transferred population.

Keywords—Transfer Learning, Multiple Sources, Knowledge Transfer, Domain Adaptation, Source, Target.

I. Introduction

TRANSFER Learning (TL) is a process of transferring knowledge from one problem to another problem. The objective of TL is to speed up the process of finding the solution to the new problem. The mission of TL is to discover and transfer the trained data. This data may be adapted from multiple source problems.

TL is considered one of the ML techniques that help some ML algorithms find solutions to problems more easily. TL usually interacts with two problems: The first one is the source problem (S), and the other problem is the target problem (T). TL transfers the knowledge from the S problem to the T problem. In some cases, the S problem may not be presented, but the knowledge of solving the S problem is stored, and we can transfer it to the T problem [1].

The comprehensive process of TL mimics how humans think. For example, humans learn from their life how to adapt and solve problems or obstacles. They build on their experience from what they have faced and also what others have shared with them. Some people may listen to a lot of advice to understand and think about how to overcome his obstacles. Many people consider this behavior a wisdom behavior.

In this study, we adopt the following real-life situation. Students go to school and learn different subjects. They learn and practice what they have learned in class to pass and have good grades. For example, they apply what they learned in mathematics classes in physics classes. We would like to study how the possibility of transferring knowledge from several source problems to the target problem. Also, this study can be applied to different situations: for example, having two robots and each one has been trained to drive in two different areas. Robot A was trained to drive and discover the sand area. Robot B was trained to drive and discover the mountain area. Having a third Robot that combines knowledge from Robot A and Robot B may save us money, training time, and effort.

To address our goal, we modeled the TL process using Genetic Algorithm (GA) [2], [3]. The GA was used as the model solver for the S and the T problems. We created five different source problems, and the model solved all of them first. Then we constructed the transferred population using the knowledge from all of these final solutions. We proposed using the Multi Bowl Transfer Population (MBTP) method to generate the transferred population (for more details, see Section III). After that, the model transferred the constructed population to the target problems. The model solved the target problems using the constructed transferred population. To make our study cover different life situations, we tested four different target problems; each one describes different approach (for more information see Section III).

Our study answers the following questions:

- **Q1.** Can we transfer knowledge from multiple source problems?
- **Q2.** How good is sampling the transferred population from multiple source problems?
- **Q3.** Can we solve a target problem that combined knowledge from two different source problems?

This paper is organized as follows: Section II explains the background of this study. Section III is the method we used in our study. Section IV is the experiment and Section V is the discussion, followed by the conclusion and acknowledgment sections.

II. Background

Liu and Wang [4] implemented TL in Dynamic Multi-Objective Optimization Algorithms (DMO). They used TL to improve the initial population prediction for the target problem. They proposed an algorithm called TPS-DMOEA that contains the following steps:

1) Select the transferred population using the Population Prediction Strategy (PPS).
2) Modify the transferred population from the first step using Transfer Component Analysis (TCA).

The authors evaluated the proposed algorithm using 10 different problems. The results showed the TPS-DMOEA algorithm overcame the existing methods of DMO.

Tami Alghamdi is with University of Idaho, USA (e-mail: algh2827@vandals.uidaho.edu).

Mendes et al. [5] implemented the TL to enhance CNN performance. They proposed a new method called Many Layer Transfer Learning Genetic Algorithm (MLTGA). They claimed their method can help the medical doctors to explore Pneumonia disease in early stages. The method built the Pneumonia's classifier model by transferring the well-trained layers from previous CNN. The results showed that the proposed algorithm was accurate with 2% higher than other GA methods.

Ardeh et al. [6] studied the uncertain capacitated arc routing problem. This problem simulates an environment that has an undirected graph that connects tasks together, and the vehicles must serve these tasks. The best solution is to find the minimum cost that serves most tasks. The Genetic Programming Hyper Heuristic (GPHH) method is used to solve the previous problem. The authors enhanced the performance of the GPHH method using the TL. The proposed method discovers and removes the duplicated individuals from the transferred population. Also, the method maintains diversity that may be affected by the removing process. The experiment showed this approach overcomes state-of-the-art genetic programming with TL methods.

Chen and Liu [7] studied the Bi-Level Optimization Problems (BLOP). This problem is complex and is considered as nondeterministic polynomial (NP) type of problem. Typically, this problem deals with two levels of optimization. The low-level optimization controls the high-level optimization. Studies show this problem can be solved by using the Covariance Matrix Adaptation Evolution Strategy (CMA-ES). The authors used the TL strategy to improve the performance of the CMA-ES strategy. Their strategy is composed of two steps. First, they restricted the search process to neighboring lower levels. Then they selected the transferred feature using the learning rate. The results showed the proposed algorithm improves the performance and efficiency compared to other CMA-ES enhanced methods.

III. Method

Our model employed the Genetic Algorithm (GA) as the solver unit. The GA allows us to know what information we have about solving a problem by analyzing the final solution of the problem. The GA generates a set of potential candidate solutions to the problem and searches among them to find the optimal's solution to the problem. The differences of the problems can be managed by controlling the fitness and the gene representations. The model blends the TL with GA by transferring the knowledge that has been learned to solve the S problem to the T problem. Our model counts how many generations the T solver took to find the solution to the T problem using the transferred population.

Our study looked at different perspectives in which the T problems require the T solver to search for different solutions. Our fitness function counted to learn new things. By changing the fitness value of the T problem or the gene representation, we can discover different approaches.

For this study, we have developed five different source problems (3) and four different target problems (4). We consider cases where the target solver must learn new things, protect the knowledge the transferred population has, or forget some knowledge. We consider whether these cases required the solver to generate many generations to find the optimal solution. The number of generations reveals the efficiency of the transferred population. In this manner, we can study how effective our model is at transferring knowledge from multiple source problems.

The model starts by solving each one of the S problems using a random initialized population and stores the final solution of each problem separately. Then we construct the transferred population using the final solutions of all source problems that we stored (see Section IV). After that, the model transfers the transferred population to the T solver to find the solution to the T problem. The model counts how many generations the T solver took to find the final solution to the T problem.

In this study, we have generated four different T problems. Our model can study different cases and approaches. For example, what is the effect of having the system initiate a random T problem and ask the T solver to solve the problem? What is the effect of having a target problem requires combined knowledge from different S problems? What is the effect or the benefit of having a T problem simulate one of the S problems? What is the effect of having a T problem that requires partial data from different S problems? We considered if the T problems will require the T solver to generate many generations to find the solution to the T problem. We also considered if the transferred population is able to solve this type of problem easily or if the transferred knowledge is missing and not able to lead the T solver to solve the problem.

We have used the generational GA. The GA will deal with a population consisting of a fixed number of individuals. Each individual consists of 40 bits or genes long. For the selection operation of the GA, the tournament selection is enabled. In this selection, selected randomly individuals are selected and they have the right to produce the new generation. In this study, the tournament size is 3. The details of the GA arguments are specified in Table I.

TABLE I
PARAMETERS OF THE GENERATIONAL GENETIC ALGORITHM (GA)

Genetic Parameter	Value
GA Type	Generational
Chromosome length	40
Population size	100
Mutation rate (per bit)	0.1
Crossover rate	0.05
Type of crossover	Uniform crossover
Tournament Size	3

Fitness Function is a function that the GA used to evaluate each individual. This function yields a value that describes how close the evaluated individual is to solving the problem. For our study, we have used a fitness function that is defined on a number of bits or genes and divided them into many subfunctions. The following is the fitness function (1):

$$f(x) = \sum_{i=0}^{m} a_i \; g_i(x[i * s, \; (i+1) * s - 1]) \tag{1}$$

where (x) is the individual. g_i is a subfunction, and it is defined over s bits subfunction. We have m subfunctions and s number of bits for each subfunction. The n is the total individual length. $n = m * s$ ($n = 10 * 4 = 40$). The a_i is [0,1] is the bit position of the x.

For our study, a_i value is fixed to [0,1]. We have chosen to deal with this type of problem because we want to show the importance of each crossbanding subfunction. For example, if the $a_i = 1$, that means the crossbanding subfunction is important and we have to solve it to achieve the solution to the problem. On the other hand, if $a_i = 0$, that means the crossbanding subfunction is not important and we do not have to find a solution to this subfunction. In other words, the only important subfunctions are the ones that crossband to 1.

We have 10 subfunctions that describe each individual. Each subfunction consists of four bits. Our model must solve each subfunction to evaluate each individual. To solve each subfunction we have implemented the deceptive function. This function is a type misleading subfunction. In general, this function shows it is improving as there is a zero in its argument, but the best solution is when all arguments or (subfunction's bits) are one. The following is the deceptive function (2):

$$g(b) = \begin{cases} s & \mathrm{bc}(b) = s \\ s - 1 - \mathrm{bc}(b) & \text{otherwise} \end{cases} \tag{2}$$

where $bc(b)$ is the bit counter function. This function evaluates each subfunction. The best answer is when all bit of the subfunction are equal to 1.

For the feasibility of our study, we created five different source problems. We combined the transferred population from the final solution of these source problems. We developed four different target problems. The model transferred the combined population to the target solver. The model dealt with each one of the T problems separately. Our model created four copies of the transferred population and solve each T problem individually. The following are the source problems:

$$\vec{a_{s1}} = (1,\ 1,\ 1,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0) \tag{3a}$$
$$\vec{a_{s2}} = (0,\ 0,\ 0,\ 1,\ 1,\ 1,\ 0,\ 0,\ 0,\ 0) \tag{3b}$$
$$\vec{a_{s3}} = (0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 1,\ 1,\ 1) \tag{3c}$$
$$\vec{a_{s4}} = (0,\ 1,\ 1,\ 1,\ 1,\ 0,\ 0,\ 0,\ 0,\ 0) \tag{3d}$$
$$\vec{a_{s5}} = (0,\ 0,\ 0,\ 0,\ 0,\ 1,\ 1,\ 1,\ 0,\ 0) \tag{3e}$$

The following are the target problems. The first target problem (4a) has been chosen by the system.

$$\vec{a_{t1}} = (1,\ 1,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0,\ 1,\ 1) \tag{4a}$$
$$\vec{a_{t2}} = (1,\ 1,\ 1,\ 0,\ 0,\ 0,\ 0,\ 1,\ 1,\ 1) \tag{4b}$$
$$\vec{a_{t3}} = (0,\ 1,\ 1,\ 1,\ 1,\ 0,\ 0,\ 0,\ 0,\ 0) \tag{4c}$$
$$\vec{a_{t4}} = (1,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0) \tag{4d}$$

As we can see, each one of the target problems discovers a different approach. For example:

- First T Problem (4a): GA(T-Random) random target problem. This problem will be initialized randomly by the system.
- Second T Problem (4b): GA(T-S1 & S3) combination of two source problems. This problem consists of a collection of the first S problem (3a) and the third S problem (3c).
- Third T problem (4c): GA(T-S4) a simulation of the fourth S problem (3d).
- Fouth T problem (4d): GA(T-S1,S2,S5) a partial matching problem. This problem is a partial matching problem of the first, second, and fifth S problems (3a), (3b), and (3e).

We hypothesize that we can transfer knowledge from multiple source problems to a target problem and this behavior will improve the GA performance. The TL can combine the knowledge from different problems, and this combination will prevent knowledge from being losing. Also, this combination will add an amount of diversity to the transferred population. The diversity will increase the search space of the GA. The GA performance can be measured by the number of generations the GA must generate to find the solution to the problem.

IV. Experiment

Our experiment looked at cases where the model samples the transferred population from multiple S problems and transfer them to the T solver. The target solver must use this population to find the solution to the T problem. We experimented with four different target problems. Each one of these problems cover a different approach.

In our experiment, we have five source problems denoted as $(S1, S2, S3, S4,$ and $S5)$ (see Fig. 1). The source solver was denoted as (GA), the solver used a randomly initialized population to solve each one of the source problems. The model solved and stored each problem individually. After the model finished solving all source problems, the model constructed the transferred population using the final solutions of all source problems. Then the model used the control unit population, a copy of each source's final population, and the constructed transferred population (MBTP) to solve the target problems. Each one of the target problems was solved individually. The model counted how many generations the target solver took to find the solution to each one of the target problems. This experiment ran for 50 iterations. For comparison purposes, we used the first initialized population (control unit) the final solutions of the source problems and the MBTP population. We compared the difference between these strategies. Fig. 1 represents the steps that we used in our experiment.

Fig. 1 shows experiment diagram. We have five different source problems and four different target problems. Our model solves these problems and stores the final solution of each problem individually. Then the transferred population is constructed using the MBTP method. After that, the model will

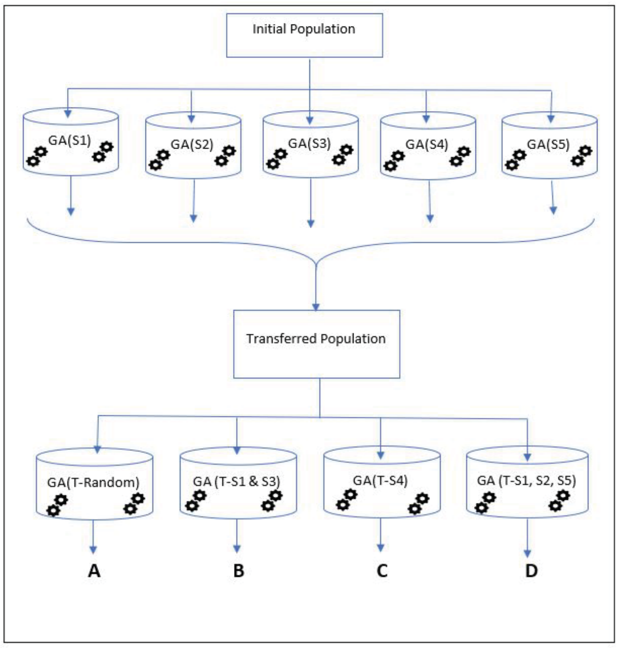

Fig. 1 Experiment Diagram, transferred population combined from all S problems

use the constructed population which is then the transferred population to solve each one of the target problems. The model counts how many generations the target solver used to find the solution to each one of the target problems.

Multi Bowl Transferred Population (MBTP) is the transferred population method we constructed for this study. In this study, we are dealing with five different source problems and we want to combine the knowledge of solving all S problems. We used 20% of each final solution of the S problems as follows:

- Top-10%: transferring the 10% of the top final solution of each S problem to the transferred population.
- Best-10% coping the best individual of each S problem 10% of the transferred population.

After we used the MBTP method to construct the transferred population, we transferred the final population to the T solver to solve the T problems.

A. First Target Problem

First target problem (4a) is the random problem. This problem was testing the random problem that was initialized by the system. The target solver must find a solution to this problem. Fig. 2 represents how many generations the target solver used to solve this problem.

Fig. 2 represents how many generations the target solver took to find a solution to the first target problem (4a). The fourth and MBTP populations show fewer number of generations compared to other populations. The control unit population represents the first random population. The target solver takes a large number of generations using the control unit population to solve this problem.

Table II is the pairwise Mann-Whitney U test. This test shows the significance (p-value) of using each type of

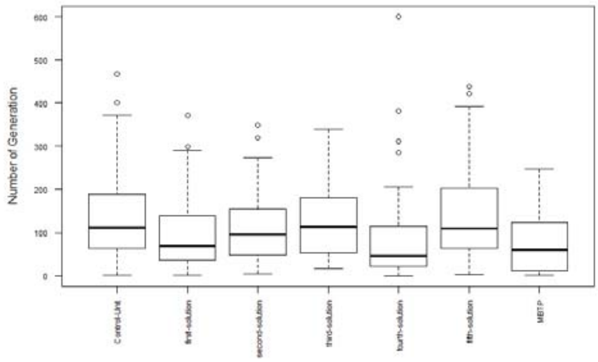

Fig. 2 Generations used by the model to solve problem (4a)

population to solve the problem. The last row of this table shows the average number of generations for each strategy.

B. Second Target Problem

Second target problem (4b) is the combination problem. This problem tests the ability of the model to solve a problem that combines knowledge from two different source problems. This problem combines knowledge from S problem (3a) and S problem (3c). The target solver must use the transferred populations to find a solution to this problem.

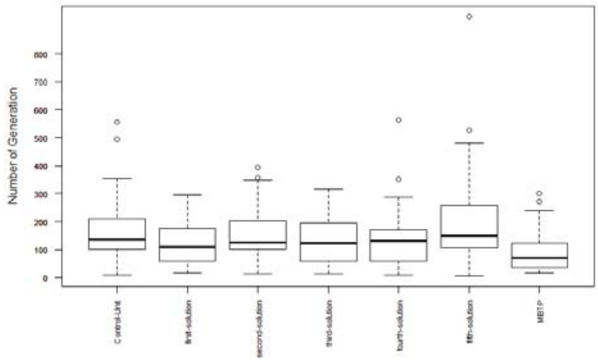

Fig. 3 Generations used by the model to solve problem (4b)

Fig. 3 represents how many generations the T solver spends to find the solution to the problem. The MBTP population shows fewer number of generations compared to the other population. The final solution of the fifth problem spends a large number of generations.

Table III shows the pairwise Mann-Whitney U test results of the number of generations. This test shows the p-value of each sampled population. The last row of this table shows the average number of generations for each strategy.

C. Third Target Problem

Third target problem (4c) is the simulation problem. This problem is exactly the same as the fourth source problem (3d). The aim of this problem was to test if the knowledge stored in

TABLE II
THE PAIRWISE MANN-WHITNEY U TEST P-VALUE FOR SOLVING THE FIRST TARGET PROBLEM (4A)

	control-unit	fifth-solution	first-solution	fourth-solution	MBTP	second-solution
fifth-solution	0.928	-	-	-	-	-
first-solution	0.018	0.021	-	-	-	-
fourth-solution	0.0005	0.0007	0.156	-	-	-
MBTP	0.0004	0.0004	0.106	0.764	-	-
second-solution	0.156	0.194	0.273	0.018	0.011	-
third-solution	0.475	0.541	0.085	0.002	0.0009	0.404
Mean	141.34	143.04	99.08	87.7	73.68	109.1

The third-solution mean is 120.56.

TABLE III
THE PAIRWISE MANN-WHITNEY U TEST P-VALUE FOR SOLVING THE SECOND TARGET PROBLEM (4B)

	control-unit	fifth-solution	first-solution	fourth-solution	MBTP	second-solution
fifth-solution	0.612	-	-	-	-	-
first-solution	0.052	0.011	-	-	-	-
fourth-solution	0.059	0.034	0.890	-	-	-
MBTP	**7**.4 **E**-06	**1**.4 **E**-06	0.007	0.003	-	-
second-solution	0.475	0.244	0.165	0.339	**5**.0 **E**-05	-
third-solution	0.077	0.019	0.814	0.893	**0.003**	0.238
Mean	168.74	196	127.52	135.56	87.82	150.64

The third-solution mean is 130.3.

TABLE IV
THE PAIRWISE MANN-WHITNEY U TEST P-VALUE FOR SOLVING THE THIRD TARGET PROBLEM (4C)

	control-unit	fifth-solution	first-solution	fourth-solution	MBTP	second-solution
fifth-solution	0.107	-	-	-	-	-
first-solution	0.375	0.088	-	-	-	-
fourth-solution	<2 **E**-16	<2 **E**-16	<2 **E**-16	-	-	-
MBTP	<2 **E**-16	<2 **E**-16	<2 **E**-16	-	-	-
second-solution	0.902	0.057	0.415	<2 **E**-16	<2 **E**-16	-
third-solution	0.137	0.745	0.017	<2 **E**-16	<2 **E**-16	0.104
Mean	134.4	150.28	88.64	0	0	104

The third-solution mean is 132.2.

the transferred population solves the problem easily or if the model can improve the solution to this problem. The model must find the solution to this problem.

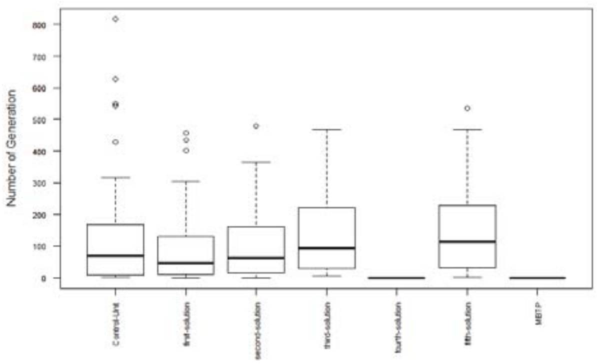

Fig. 4 Generations used by the model to solve problem (4c)

Fig. 4 represents how many generations the T solver takes to find a solution to the fourth T problem. This figure shows that the target solver spends 0 generations using the final solution of the fourth source problem (3d) and the MBTP populations. This is because these two populations already have the solution for this problem.

Table IV is the pairwise Mann-Whitney U test. This test shows the significant p-values for each transferred population. The last row represents the average number of generations for each population.

D. Fourth Target Problem

Fourth target problem (4d) is a partial matching problem. This problem has knowledge from source problems one, two, and five (3a), (3b), and (3e). This problem aims to test the model ability to solve a problem that required knowledge from multiple different parts of the final solutions of multiple source problems. The target solver must use parts of the knowledge that is stored in the population to find a solution to this problem.

Fig. 5 represents how many generations the T solver spends to find the solution to this problem. The MBTP population shows fewer number of generations compared to the other population type.

Table V is the pairwise Mann-Whitney U test. This table shows the p-values of the significance level of the number of generations for each population type. The last row shows the average number of generations the T took to find the solution to the problem.

TABLE V
THE PAIRWISE MANN-WHITNEY U TEST P-VALUE FOR SOLVING THE FOURTH TARGET PROBLEM 4D

	control-unit	fifth-solution	first-solution	fourth-solution	MBTP	second-solution
fifth-solution	0.108	-	-	-	-	-
first-solution	0.574	0.236	-	-	-	-
fourth-solution	0.330	0.318	0.801	-	-	-
MBTP	0.006	0.360	0.031	0.045	-	-
second-solution	0.055	0.955	0.201	0.323	0.491	-
third-solution	0.183	0.616	0.400	0.614	0.168	0.552
Mean	261.5	192.98	234.62	198.7	97.52	186.56

The third-solution mean value is 178.3.

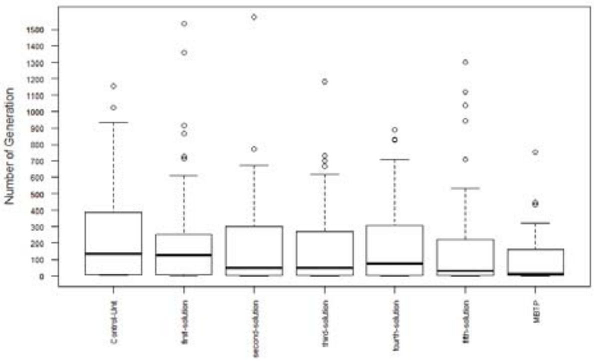

Fig. 5 Generations used by the model to solve problem (4d)

V. DISCUSSION

This study showed that transferring knowledge from multiple source problems is possible and may help algorithm designers to improve the GA performance. Knowledge from multiple sources can be combined together in one population and solve more advanced problems. For example, our study gathers knowledge from five different source problems and applied it to solve the target problem.

By analyzing the transferred population that contains the combined knowledge, we found that the process of combining knowledge from multiple source problems added some diversity to the transferred population. Also, the combining process protected the old knowledge from loss. According to other studies such as [8] and [9], old knowledge and diversity are two important components that must be available in the transferred population.

The MBTP method constructed the transferred population using knowledge from final solutions of multiple source problems. This method follows the fashion of exploiting and exploring. The exploitation was enhanced by using the 10% copy of the best individual of each source problem, and the exploration was enhanced by transferring the top 10% of each source problem's final solution.

In this study, we have used diversity from the source problems that we had already used to solve our problems. We transferred the top 10% of each final solution to the transferred population to enhance the population diversity. We feel this diversity did not cover all possibilities since it was used to solve other problems. In the future, we must investigate what happens if we use a totally random individual as population diversity.

VI. CONCLUSION

We studied how to transfer knowledge from multiple source problems. We proposed the MBTP method, which samples the transferred population using knowledge and diversity from solutions of the solved source problems. We experimented with five different source problems. We constructed the transferred population and transferred it to the target solver to solve the target problems.

We studied four different approaches of target problems. These approaches cover some real-life scenarios. For example, if we have two robots that had been trained to drive in two different environments, we can combine their knowledge into one robot that can serve in both environments.

Transfer Learning can combine data from multiple sources in one population. Our proposed method may help the GA task to solve more advanced problems or at least protect the knowledge from loss. The experiment and results show gathering knowledge from multiple sources improves the GA performance compared to starting from scratch.

ACKNOWLEDGMENT

This study is fully supported by Al Baha University, Al Baha, Saudi Arabia, and (SACM) Saudi Arabian Cultural Mission. The author would like to thank supervisor Prof. Terence Soule for his support and advice in this study.

REFERENCES

[1] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.
[2] A. E. Eiben, J. E. Smith *et al.*, *Introduction to evolutionary computing*. Springer, 2003, vol. 53.
[3] M. Mitchell, *An introduction to genetic algorithms*. MIT press, 1998.
[4] Z. Liu and H. Wang, "Improved population prediction strategy for dynamic multi-objective optimization algorithms using transfer learning," in *2021 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2021, pp. 103–110.
[5] R. de Lima Mendes, A. H. da Silva Alves, M. de Souza Gomes, P. L. L. Bertarini, and L. R. do Amaral, "Many layer transfer learning genetic algorithm (mltlga): a new evolutionary transfer learning approach applied to pneumonia classification," in *2021 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2021, pp. 2476–2482.
[6] M. A. Ardeh, Y. Mei, and M. Zhang, "Surrogate-assisted genetic programming with diverse transfer for the uncertain capacitated arc routing problem," in *2021 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2021, pp. 628–635.
[7] L. Chen and H.-L. Liu, "Transfer learning based evolutionary algorithm for bi-level optimization problems," in *2021 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2021, pp. 1643–1647.

[8] A. Gupta and Y.-S. Ong, "Genetic transfer or population diversification? deciphering the secret ingredients of evolutionary multitask optimization," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2016, pp. 1–7.

[9] T. Alghamdi and R. B. Heckendorn, "An evolutionary computation based model for testing transfer learning strategies," in *2021 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2021, pp. 1380–1389.