International Journal of Innovative Computing, Information and Control
Volume **8**, Number **10(A)**, October **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 6799–6809

TRANSFER LEARNING IN VEHICLE ROUTING PROBLEM FOR RAPID ADAPTATION

Barış Koçer and Ahmet Arslan

Department of Computer Engineering
Faculty of Engineering and Architecture
Alaadin Keykubat Kampüsü
Selcuk University
Selçuklu, Konya, Turkey
{ bariskocer; ahmetarslan }@selcuk.edu.tr

Received June 2011; revised December 2011

Abstract. *Vehicle routing problem is a transportation optimization problem in transporting stuffs from depot(s) to receivers via vehicle(s) having limited capacity. There are a lot of different problem types in VRP literature with different problem parameters. No VRP method uses past solutions to solve current problems more quickly and to find better solution with less computation. Instead, most of the methods in literature evaluate the changes as a new problem and try to solve the new problem using specialized heuristics. We developed a method which uses past vehicle routes to make new routes quickly for frequently changing conditions, and we achieved good performance improvements over classical methods.*

Keywords: Vehicle routing problem, Transfer learning, Genetic algorithms

1. **Introduction.** In this study, we developed a method which uses a modified version of genetic transfer learning [13]. Genetic transfer learning has been developed to take advantage of transfer learning techniques in optimization problems. The proposed method uses some useful parts of the old plan to create a new plan more quickly and better than a plan which is made without using information from the old one when a new routing plan is urgent. Another advantage of the proposed method is its adaptation ability to more than one problem type unlike the current methods.

In this section, the VRP problem and transfer learning which are the bases of the proposed method are separately defined. Section 2 contains literature review of the VRP problem and transfer learning. In Section 3, classical methods and the proposed method are introduced. Section 4 contains numerical and graphical performance comparisons and last section makes conclusions and contains further potential improvements for the proposed method.

1.1. **The vehicle routing problem.** VRP can be summarized as deciding on a routing plan for vehicles to transport stuffs from depot(s) to receivers via limited capacity vehicle(s). In a scientific way VRP is a graph-theoretic problem. Let G = (V, A) is a complete graph where V = $\{0, 1, 2, \ldots, n\}$ is a set of vertices, in which each vertice corresponds to a customer with a pre-determined order o_j; and A is an arc between customers with a travelling cost d_{ij} between customer i and customer j. In classical VRP, the travelling cost between two customers in both directions are the same, i.e., $d_{ij} = d_{ji}$ and there is only one depot which is V_0, but in real life VRP problems may be very complex. For example, depot count and capacity of vehicles may vary and even the number of points to travel may not be pre-determined and may vary after a vehicle begins its travel. A taxonomic review for these scenarios is presented by Eksioglu et al. [1]. We are especially interested in changing travelling times and changing customer demands because these are very frequent situations in big VRP problems. For a classical VRP problem, travelling times between each point and customer orders are pre-determined and it is assumed that these parameters are fixed. But in real life travelling time may vary via rush hours and customers may add or delete their orders according to their needs. With the help of new emerging communication systems, vehicles can be informed about every change and all changing situations can be sent to vehicles via communication systems like mobile Internet which may cause changes in their routes. It is very important to adapt every changing condition as quickly as possible to minimize total travelling costs.

1.2. **Transfer learning.** Traditional machine learning techniques use training data to train an expert system to make a prediction for unseen data. The main assumption in machine learning is that training data and unseen data have the same distribution. However, in real life this assumption might not work all the time. Data distributions and environmental conditions are subject to change and sometimes training data can easily become outdated. Traditional machine learning techniques need new training data to adapt for every changed condition. But sometimes it is very difficult or costly to acquire new training data when need arises since the current training data are outdated. In such a case, if the machine learner system can use past experiences like human intelligence does, then predictive accuracy may increase with less training data for newly changed conditions. For example, a person who knows how to play ping-pong can learn how to play tennis more easily than one who has never played ping-pong. A similar view concerning this situation is explained with a self learner system by Eguchi et al. [27].

Transfer learning can be summarized as an approach to increasing performance for learning tasks which have insufficient resources by using resources of a related task. In transfer learning there are two kinds of tasks. One of them is the target task which has insufficient or no resources which can be used in traditional machine learning. The other kind of task is the source task which has enough resource related or similar to the target task. The aim of transfer learning is improving performance of the target task by using all available resources of the source task and target task. For example, in an indoor Wi-Fi localization task, an office splits into fixed width and height cells and the strengths of Wi-Fi spots are measured on some of these cells. These measurements are used as training data to build a full mapping from Wi-Fi spot signal strength to cell number, on which the receiver exists. So this source task can be trained by using a lot of measurements of signal strengths on various cells. After system is trained well, if the signal strengths of the Wi-Fi spots are changed due to temperature, humidity, moving objects, etc. or the receiver's hardware is altered, should all training be done again with new training data? The indoor environment may be so large that re-measurement may necessitate huge human effort. This is an example for outdated training data, so new situation can be assumed as the target task. The target task can adapt to this new situation by a little training data with the help of past knowledge obtained from source tasks. Differences between traditional machine learning and transfer learning are illustrated in Figures 1(a) and 1(b).

Although transfer learning is a good method for improving performance of a target task, it has some problems which should be solved before the knowledge transfer. Main difficulties which are met in transfer learning can be summarized in four categories below.

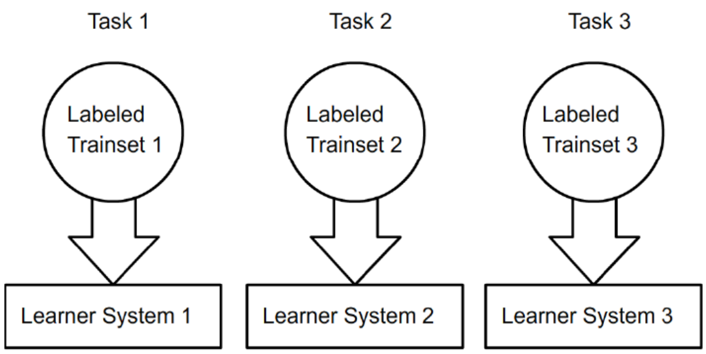
1. Determining relatedness of source and target tasks: If unrelated source task is selected for transfer it may decrease the performance of target task. This situation is named as "negative transfer".

2. Determining the amount of knowledge to be transferred from the source task to target task: If too little knowledge is transferred it will not have any effect on target task, but if too much knowledge is transferred it may result in "negative transfer".

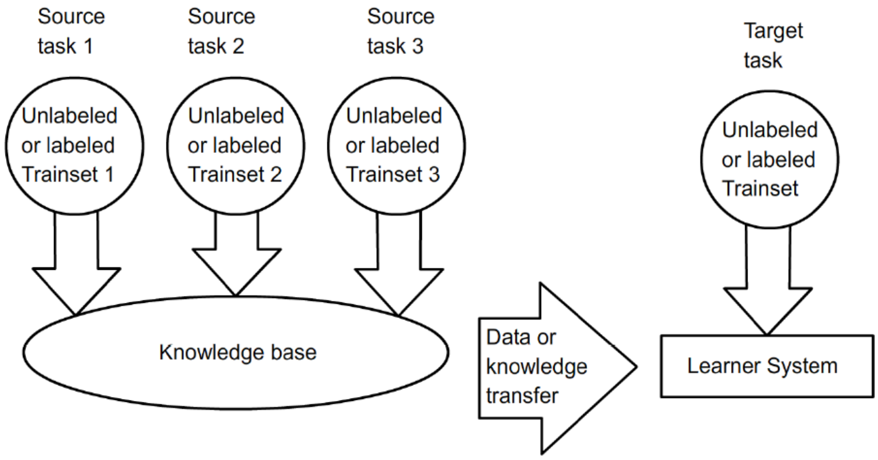
3. How to transfer the knowledge from source task to target task: This is maybe the biggest problem in transfer learning. There are a lot of machine learning methods modified for transfer learning.

4. How to store acquired knowledge in order to use on similar target tasks in future.

One who wants to use transfer learning efficiently has to solve the problems above. A very good comprehensive survey for transfer learning is prepared by Pan and Yang [2].



(a)



(b)

FIGURE 1. (a) Traditional machine learning methods' working principle, (b) transfer learning working principle

2. Related Studies.

2.1. **Transfer learning.** Knowledge transfer is established in different ways. Modifying a learning method for knowledge transfer is a common practice. One of the frequently modified machine learning methods is reinforcement learning. Reinforcement learning is adapted to transfer learning for skill transfer [3], action schema transfer [4] and control knowledge transfer [5]. One of the best samples for transfer learning is indoor Wi-Fi localization task which is a training a model in an indoor environment which is split into fixed cells to determine where is the receiver. This task is simply predicting cell coordinate of a receiver using signal strength of Wi-Fi spots. Transfer learning becomes essential when labeled data for indoor Wi-Fi localization task become out of date easily due to new obstacles or reflection in time [6] or hardware of the receiver or access points is changed [7]. In a classical machine learning approach new training data should be obtained again and again for every changing condition but transfer learning methods can be applied to adapt old training data to new situation. Transfer learning can also be used for obtaining good performance with sparse labeled data in Wi-Fi localization task. For example, training data may be obtained only for a little part of a very big indoor environment and it is very difficult to obtain full map of the environment [9], so when transfer learning methods are applied, the amount of labeled data needed to build localization model is significantly reduced. Transfer learning methods are also applied for text categorization tasks. Among the previous studies focusing on text categorization with transfer learning are Dai et al.'s study [10] which uses expectation maximization based on Naïve Bayes classifier and another one by Eaton et al. [11], which uses graph based transferability measurement and extracts transfer parameters from source tasks. In [12], Dai et al. proposed a text categorization method which is a modified version of AdaBoost algorithm to leverage old labeled data with the help of a little newly labeled data to build an accurate classification model. In this study, we have developed a genetic algorithm based transfer learning method and it is the first example for genetic algorithms usage in transfer learning [13].

2.2. **Vehicle routing problem.** Travelling salesman problem is an NP hard optimization problem. VRP is a scientific case, which is a much more complex form of the TSP. The first paper about the VRP was by the Dantzig et al. [14]. The term "Vehicle routing" emerged in the paper by Golden et al., [15]. A similar term like "Fleet routing" [16], "transportation network design" [17,18] is used in following years. First VRP problems were all deterministic cases, i.e., customer demands, vehicle capacities and counts and travelling cost were pre-determined but in practice some of the problem parameters were probabilistic. Probabilistic parameters were first added to VRP by Golden and Stewart [19] and other studies can be referred to as Laporte and Nobert [20], Solomon [21].

In this work, we especially studied changing travelling costs, i.e., changing travelling times between two customers by virtue of traffic density or changing customer demands by adding or deleting the orders. Previous works in this area can be given in the main category of "quality of information" and under the sub-category "Unknown (Realtime)" of taxonomy presented in [1]. These works took interest in solving problems which appeared from changing conditions like changing travelling times [22-24] or customer demands [25]. The common point of these works is that vehicles are informed for changed conditions and the systems on vehicles are expected to make new routing plans according to new conditions.

3. **Transferring Route Plan.**

3.1. **Genetic algorithms for VRP.** We used genetic algorithms to calculate travelling costs. Travelling sequence of the customers is coded in genes and the best combination which minimizes travelling cost is selected as optimum solution. We use 100 customers, 1 vehicle without capacity limit and symmetric travelling times. We used 100 as the population count and different maximum generation counts to see the performance effects of the proposed method. Since traditional crossover operation may cause unsuitable solutions, order crossover operator proposed by Davis [26] and illustrated in Figure 2 is used to generate new offspring.

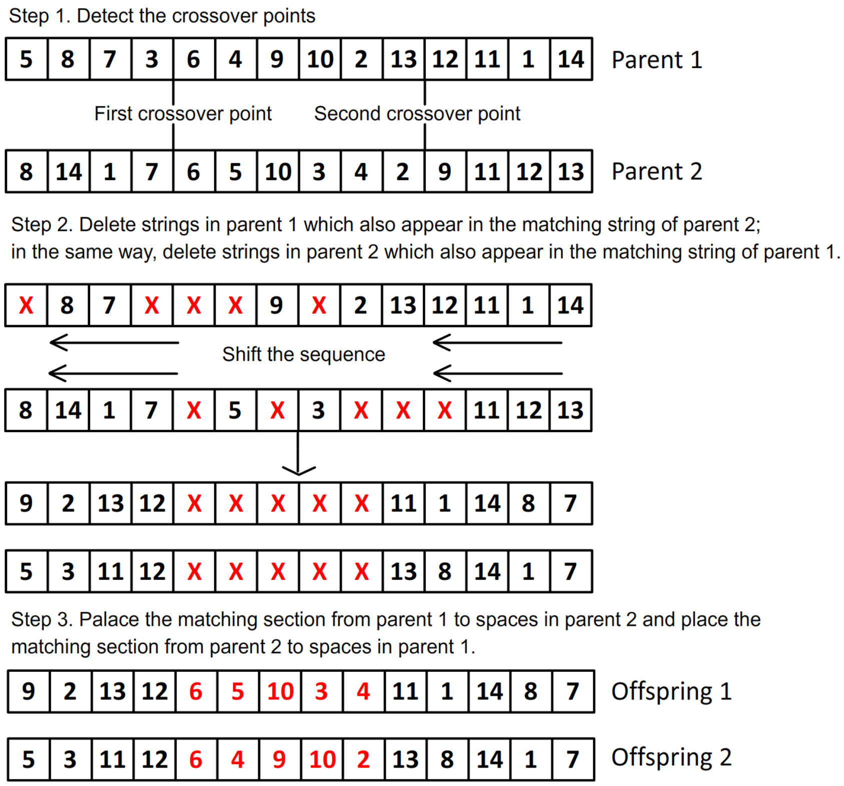


Figure 2. Order crossover operator by Davis [26]

3.2. **Route transferring for genetic algorithms.** We have used a new version of the genetic transfer learning [13] with some modification which was made to adapt the method to VRP. Genetic transfer learning which is illustrated in Figure 3 leverages knowledge transfer capabilities of transfer learning for optimization tasks. The proposed method transfers the knowledge gathered from previously solved optimization problem to the new one. In transfer learning, source task and target task are different tasks with different training data but in genetic transfer learning, the source task is previously solved optimization task and the target task is the need for new optimizations due to a changing condition in the problem. A classical approach calculates a new route plan from scratch for every changing condition or develops a new heuristic to adapt new situation to the old route. Calculating a new route plan approach is not suitable for time crucial systems because every new calculation requires additional time and developing a new heuristic for each type of problem needs huge human effort for a wide variety of VRP problems. Vehicle counts, customer demands, travelling times, etc. are some of the sample parameters which may change. Current state of art techniques are heuristic methods which are specialized in one of the problem types above. There is not a method which can solve many of the VRP problem types. But in real life there is a need for more flexible systems which can deal with many changing conditions. In this work, we developed a method which can solve two problem types changing travelling time and changing customer demands.

In genetic algorithms every new population is generated from the best fitness valued individuals of previously generated population. What about other individuals? Lower fitness valued individuals are not considered and are not used for generating a new population but every generation is a small exploration of the whole solution space. For similar problems, these lower fitness valued individuals or even higher fitness valued ones may be a good starting point when compared with starting to search randomly from scratch.

But it is not practice to save and evaluate all the generation and so we have chosen some special individuals which can represent the whole generation well. Thus, we have chosen to save the best, middle and worst fitness valued individuals into "solution pool" for every new population. Solution pool is used for further target tasks for possible existence of suitable solutions. The first use of transfer learning in genetic algorithm was in our previous study [13]. In this work, solution pool creation method was almost the same with the method in the current study, but instead of randomly selecting some solutions from the solution pool for transferring the target task, we evaluated the all solutions for the target task and transferred only the best solution. In VRP, when some of travelling times changed, some parts of the routing plan may change, but the unchanged parts of the old plan can be reused. Thus, we developed a method based on transfer learning to use the useful part of the old plan to create a new effective plan quickly.

The details of the genetic transfer learning algorithm are illustrated in Figure 3. In this algorithm, solutions in the solution pool may be unsuitable for new conditions, i.e., in the solution there may be customers who have already been visited, so before evaluating the solution for new travelling times these customers should be eliminated. Thus, Algorithm 1 is developed and used to create new genes from the solutions in the solution pool and evaluate for the current situation.

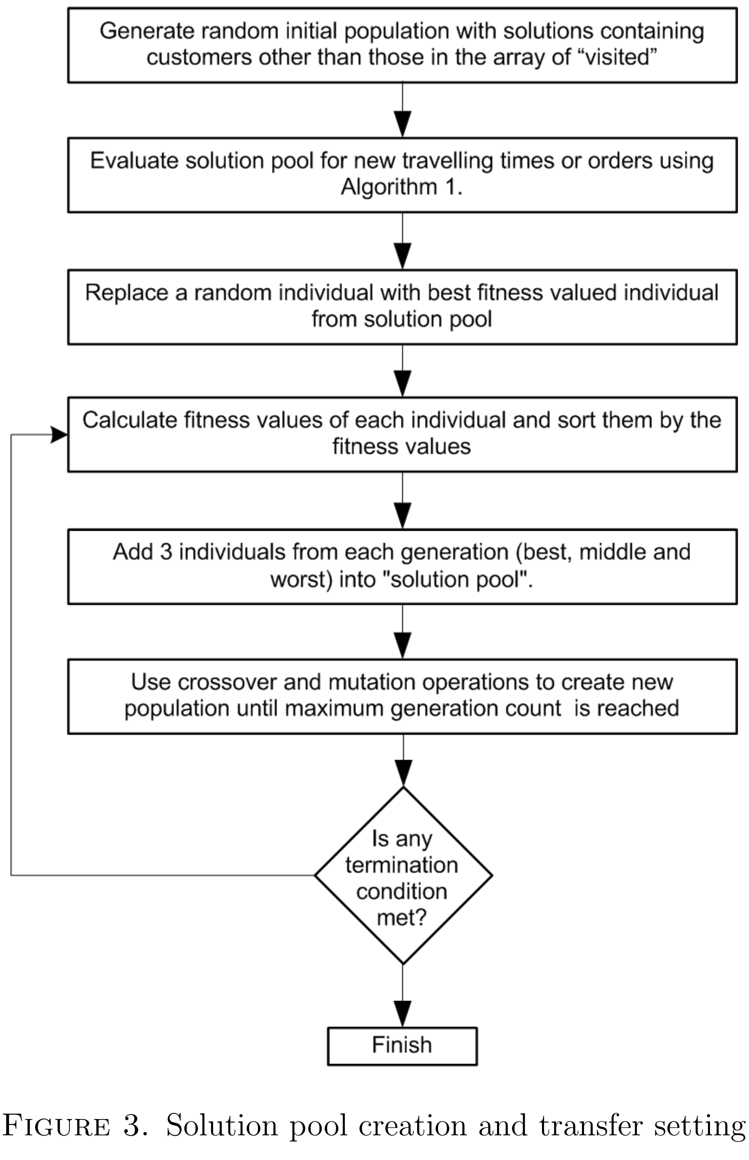


Figure 3. Solution pool creation and transfer setting

Variables:
V: Array of visited customers,
C: Count of remaining customers to be visited,
B: Customer number which the vehicle is currently on,
C_G: A counter for generating new genes,
T: Solution which is taken from solution pool and altered for current situations,
L: Gene count of T,
Foreach solution "S" in solution pool do
Set all gene of T to "−1"
Set 1.th gene of T to B
Set $C_G = 2$
For i = 0 to L do
Begin
    Read i-th gene of S and assign it to R
    If R is not in V
    begin
        Set C_G.th gene of the T as R
        $C_G = C_G + 1$
        If $C_G >= C$ then break the loop
    End
End
For i = 0 to C do
Begin
    If i-th gene value of the T is "−1"
    begin
        Generate a random customer number which is not in V and set it to i.th gene of the T
    End
end
    Calculate the fitness value of the T

Algorithm 1. Evaluating solution pool for new solution

4. **Experimental Settings and Results.**

Performance of proposed method is tested against two problem types of VRP.

4.1. **Changing travelling times.**

We tested the performance of our method against classical genetic algorithm for changing travelling times problem. Genetic algorithms are used to compare the proposed method when the travelling times are subject to change. We used only one vehicle without capacity limit and only one depot because the aim of the work is show the performance of the method under frequently changing conditions. We used 50 as the population count and used different maximum generation counts to measure performance. Elitism is used to survive the best solution to next generations, and order crossover [26] is used as a crossover operator which is described in Section 3.1.

The table of travelling times is a virtual table that shows the travelling times between two points for all combinations of the available points. For example, for a system which contains 100 points, there is $100 \times 100 - 100 = 9900$ different travelling times. To determine how often travelling times change, we used a variable "changing ratio" (CR). CR represents the ratio of changed travelling time count when a vehicle arrives to a customer on its route to total travelling time count. For the system above which has 9900 travelling times, when CR = 10%, 990 travelling times are changed each time the vehicle arrives a customer on its way. Performance results for the proposed method and for the genetic algorithms are shown from Figure 4 to Figure 8 for different CR values and maximum generation counts (MGC). Graphs are drawn from the mean of 10 independent runs. In each run travelling time table is recreated randomly. When the vehicle reaches customer it takes new travelling times information, solutions in solution pool are evaluated for new travelling times and the solution with best fitness value is transferred to initial population of the task.

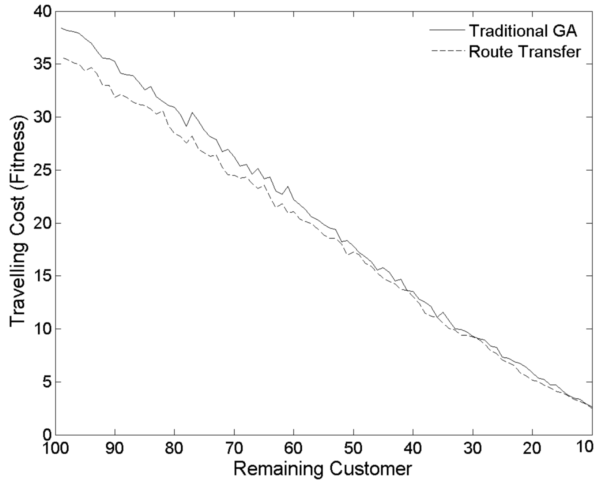


Figure 4. Performance results when changing ratio is 10%, and maximum generation count is 10

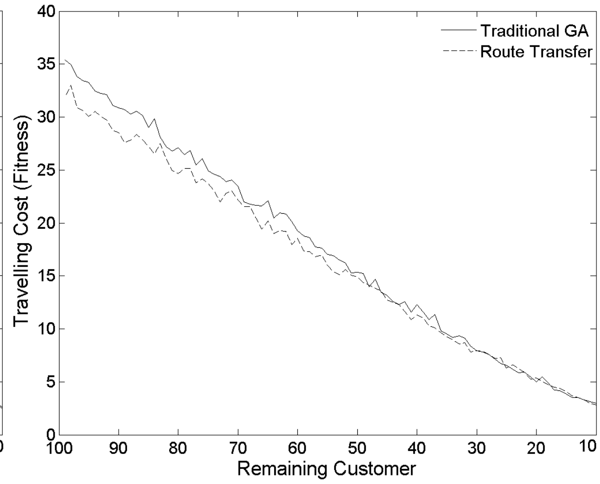


Figure 5. Performance results when changing ratio is 20%, and maximum generation count is 20

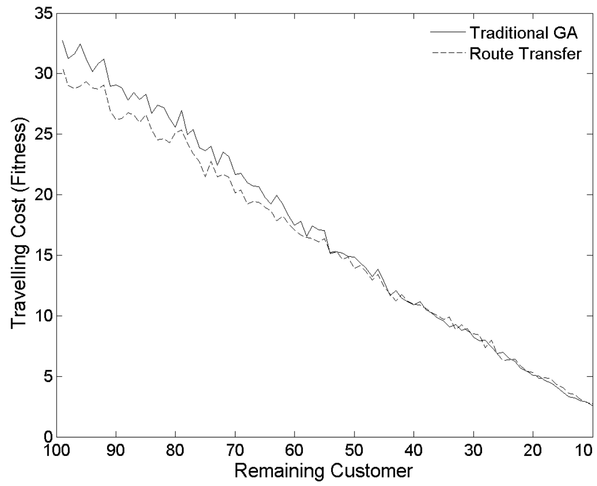


Figure 6. Performance results when changing ratio is 30%, and maximum generation count is 30

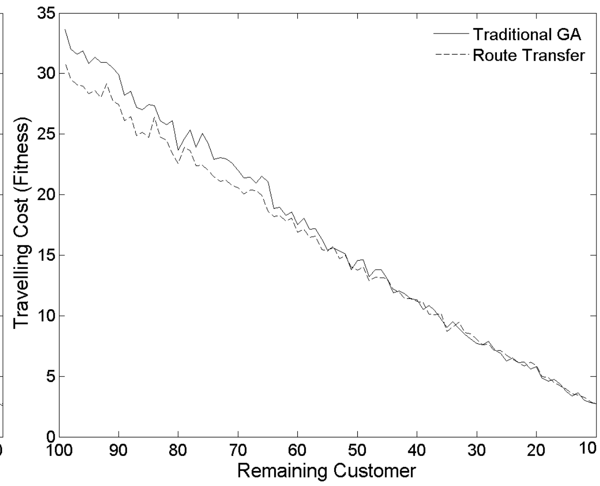


Figure 7. Performance results when changing ratio is 40%, and maximum generation count is 40

4.2. **Changing customer demands.** The proposed method is also tested under changing customer demands in comparison with genetic algorithms and classical insertion method [28]. Classical insertion method is a heuristic method which is developed for

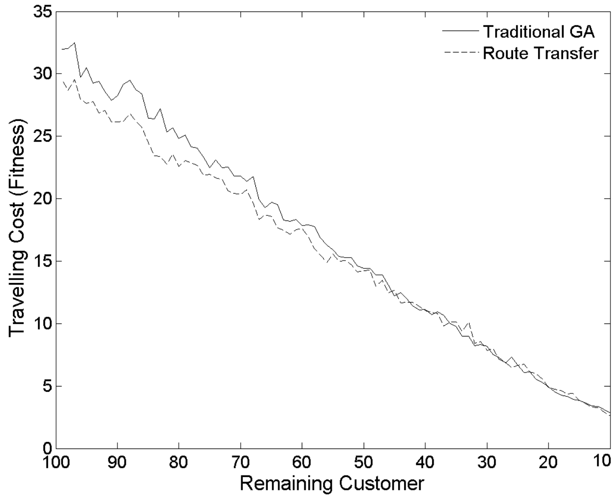


Figure 8. Performance results when changing ratio is 50%, and maximum generation count is 50

Table 1. Performance comparisons of proposed method versus genetic algorithms and classical insertion methods

MGC	***CR***	***Genetic Algorithms***	***Classical Insertion***	***Transfer Learning***	***Improvement over genetic algorithms***	***Improvement over classical insertion***
10	*5%*	*3992.202*	*3701.625*	*3294.624*	*21.17%*	*12.35%*
10	*10%*	*4172.092*	*3623.855*	*3589.309*	*16.24%*	*0.96%*
30	*5%*	*3346.831*	*3203.346*	*2723.332*	*22.89%*	*17.63%*
30	*10%*	*3502.202*	*3155.215*	*2973.407*	*17.78%*	*6.11%*
60	*5%*	*3088.731*	*3020.257*	*2560.119*	*20.65%*	*17.97%*
60	*10%*	*3261.687*	*2911.345*	*2770.731*	*17.72%*	*5.07%*
90	*5%*	*2992.391*	*2847.496*	*2437.38*	*22.77%*	*16.83%*
90	*10%*	*3184.283*	*2679.184*	*2580.074*	*23.42%*	*3.84%*
120	*5%*	*2915.873*	*2833.861*	*2409.265*	*21.03%*	*17.62%*
120	*10%*	*3053.44*	*2662.663*	*2551.446*	*19.67%*	*4.36%*
150	*5%*	*2841.548*	*2699.273*	*2363.548*	*20.22%*	*14.20%*
150	*10%*	*3029.956*	*2627.185*	*2442.793*	*24.04%*	*7.55%*
190	*5%*	*2798.275*	*2668.407*	*2254.658*	*24.11%*	*18.35%*
190	*10%*	*2913.909*	*2831.944*	*2360.951*	*23.42%*	*19.95%*

changing customer demands and has been used as a base heuristic for further methods. Insertion method is simply inserting new customer demands to suitable positions in the current solution which minimizes the total cost. First solution of the insertion method is generated by the genetic algorithms with maximum generation count taken from MGC column in Table 1. We used the same genetic algorithm setting with changing travelling time problem. We updated the pre-determined orders by adding or deleting new orders of random customers each time the vehicle arrives to a customer on its route, but travelling times between customers are left unchanged. Changing ratio of the pre-determined orders is taken from the CR column in Table 1. Performance results of the proposed method, genetic algorithms and classical insertion method are illustrated in Table 1. The results are obtained from the mean of total travelling cost values after 10 independent runs with the same parameters.

5. Conclusions.

As it is seen in changing travelling times experiments, transfer learning has yielded better than classical genetic algorithms.

In changing customer demand test, as seen in Table 1, the performance of transfer learning is better than both genetic algorithms and classical insertion method. But it can also be seen that performance improvement is huge when changing rate is 5% because a large portion of the old routes remains unchanged. It can be said that performance improvement increases when maximum generation count (MGC) gets larger. This is because the information in solution pool gets larger for every new generation.

A more important advantage of the proposed method is that it can effectively solve the main problems of transfer learning introduced in Section 1.1. This is because the proposed method can select best routing plan for new conditions via natural selection and combines it with the power of genetic algorithms. This means that calculating task relatedness, determining the amount of knowledge to be transferred and deciding on the method of knowledge transfer are done by the proposed method automatically without need for user intervention. Enabling reusability via solution pool is another advantage of the proposed method, i.e., solution pool can be used for further problems by evaluating it for new target tasks.

This work also showed that it is possible to solve more than a problem type with the same method using the knowledge transfer abilities of the transfer learning. As the result, proposed method can be used instead of classical methods when a small change needed in current routing plan and this yields quick adaptation to frequently changing conditions.

For further studies, the method can be improved to cope with different problem types and even it is possible to extend the method to handle more than one problem type at the same time by modifying solution pool evolution algorithm. Another further study can be carried out on solving the grooving solution pool problem because in every new solution, the pool gets grater and after a point it is impossible to use the pool as a knowledge source.

Acknowledgement. This study is supported by Selcuk University Scientific Research Projects Coordinatorship/Konya, Turkey. The authors would like to thank the editors and anonymous reviewers of this manuscript for their very helpful suggestions.

REFERENCES

[1] B. Eksioglu, A. V. Vural and A. Reisman, The vehicle routing problem: A taxonomic review, *Computers and Industrial Engineering*, vol.57, pp.1472-1483, 2008.

[2] S. J. Pan and Q. Yang, A survey on transfer learning, *IEEE Transactions on Knowledge and Data Engineering*, 2009.

[3] G. Konidaris and A. Barto, Building portable options: Skill transfer in reinforcement learning, *Proc. of the 20th International Joint Conference on Artifical Intelligence*, Hyderabad, India, pp.895-900, 2007.

[4] P. R. Cohen, Y. Chang and C. T. Morrison, Learning and transferring action schemas, *Proc. of the 20th International Joint Conference on Artifical Intelligence*, Hyderabad, India, 2007,

[5] S. Fernández, R. Aler and D. Borrajo, Transferring learned control-knowledge between planners, *Proc. of the 20th International Joint Conference on Artifical Intelligence*, Hyderabad, India, 2007.

[6] V. W. Zheng, Q. Yang, W. Xiang and D. Shen, Transferring localization models over time, *Proc. of the 23rd Assoc. for the Advancement of Artificial Intelligence Conf. Artificial Intelligence*, pp.1421-1426, 2008.

[7] V. W. Zheng, S. J. Pan, Q. Yang and J. J. Pan, Transferring multi-device localization models using latent multi-task learning, *Proc. of the 23rd Assoc. for the Advancement of Artificial Intelligence Conf. Artificial Intelligence*, pp.1427-1432, 2008.
[8] J. Rennie, *20 Newsgroups Data Set, Sorted by Date*, http://www.ai.mit.edu/jrennie/20Newsgroups/, 2003.
[9] S. J. Pan, D. Shen, Q. Yang and J. T. Kwok, Transferring localization models across space, *Proc. of the 23rd Assoc. for the Advancement of Artificial Intelligence Conf. Artificial Intelligence*, pp.1383-1388, 2008.
[10] W. Dai, G. Xue, Q. Yang and Y. Yu, Transferring naive Bayes classifiers for text classification, *Proc. of the 22nd Assoc. for the Advancement of Artificial Intelligence Conf. Artificial Intelligence*, pp.540-545, 2007.
[11] E. Eaton, M. DesJardins and T. Lane, Modeling transfer relationships between learning tasks for improved inductive transfer, *European Conference on Machine Learning*, 2008.
[12] W. Dai, Q. Yang, G. Xue and Y. Yu, Boosting for transfer learning, *Proc. of the 24th Int'l Conf. Machine Learning*, pp.193-200, 2007.
[13] B. Koçer and A. Arslan, Genetic transfer learning, *Expert System with Application*, vol.37, pp.6997-7002, 2010.
[14] G. Dantzig, R. Fulkerson and S. Johnson, Solution of a large-scale travelling salesman problem, *Operations Research*, vol.2, pp.393-410, 1954.
[15] B. L. Golden, T. L. Magnanti and H. Q. Nguyan, Implementing vehicle routing algorithms, *Networks*, vol.7, no.2, pp.113-148, 1972.
[16] A. Levin, Scheduling and fleet routing models for transportation systems, *Transportation Science*, vol.5, no.3, pp.232-256, 1971.
[17] A. D. O'Connor and C. A. De Wald, A sequential deletion algorithm for the design of optimal transportation networks, *The 37th National Meeting of the Operations Research Society of America, Bulletin of the Operations Research Society of America*, vol.18, no.1, 1970.
[18] L. Yang and K. Li, The railway transportation planning problem and its genetic algorithm based tabu search algorithm, *ICIC Express Letters*, vol.3, no.3(A), pp.361-366, 2009.
[19] B. L. Golden and Jr. W. Stewart, Vehicle routing with probabilistic demands, *Computer Science and Statistics, the 10th Annual Symposium on the Interface*, pp.252-259, 1978.
[20] G. Laporte and I. H. Osman, Routing problems: A bibliography, *Annals of Operations Research*, vol.61, pp.227-262, 1995.
[21] M. Solomon, Vehicle routing and scheduling with time window constraints: Models and algorithms, *Technical Report, No.83-42*, College of Business Admin., Northeastern University, 1983.
[22] B. Fleischmann, S. Gnutzmann and E. Sandvoss, Dynamic vehicle routing based on online traffic information, *Transportation Science*, vol.38, no.4, pp.420-433, 2004.
[23] S. Kim, M. E. Lewis and C. C. White, Optimal vehicle routing with real-time traffic information, *IEEE Transactions on Intelligent Transportation Systems*, vol.6, no.2, pp.178-188, 2005.
[24] M. W. P. Savelsbergh and M. Sol, DRIVE: Dynamic routing of independent vehicles, *Operations Research*, vol.46, pp.474-490, 1998.
[25] M. Gendreau, G. Laporte and R. Seguin, A tabu search heuristic for the vehicle routing problem with stochastic demand and customers, *Operations Research*, vol.44, no.3, pp.469-477, 1996.
[26] L. Davis, Job shop scheduling with genetic algorithms, *Proc. of an International Conference on Genetic Algorithms and Their Applications*, pp.136-140, 1985.
[27] K. Eguchi, S. Kurebayashi, H. Zhu, T. Inoue and F. Ueno, A self-learning support system for pupils based on a fuzzy scheme, *International Journal of Innovative Computing, Information and Control*, vol.4, no.10, pp.2441-2450, 2008.
[28] S. Roy, J. M. Rousseau, G. Lapalme and J. A. Ferland, Routing and scheduling of transportation services for disabled: Summary report, *Technical Report*, Centre de Recherche sur les Transports, Université de Montréal, 1984.