Automatic User Profiling for Intelligent Tourist Trip Personalisation

Liam Attard
University of Malta
liam.attard.18@um.edu.mt

Dr Josef Bajada
University of Malta
josef.bajada@um.edu.mt

ABSTRACT

We present a tourist itinerary recommendation algorithm that assists upcoming tourists by autonomously generating a personalised holiday plan according to their constraints. The system automatically builds a travel interest profile from the user's social media presence and recommends places tailored to the user's interests. Convolutional Neural Networks (CNN) classify the user's social media pictures into their respective travel category; Beach, Clubbing, Nature, Museums, Bars and Shopping. These determine their predominant travel interest topics. Then, two meta-heuristics, Particle Swarm Optimisation (PSO) and Genetic Algorithms (GA) use this computed travel profile to optimise a personalised plan. We evaluate the performance based on their plan quality and performance.

The system was packaged into an application that allows users to connect with their social media accounts and generate a plan for a holiday in Malta. We assessed the effectiveness of implementing automatic personalisation in a holiday planning application through in-depth semi-structured interviews by comparing a personalised and a more generic itinerary.

1. INTRODUCTION

Leisure travelling is an impactful industry whose economic importance significantly improves each year, contributing to 10.4% of the global GDP in 2019 [27]. Despite this, planning for a trip to a foreign city requires a substantial amount of time-consuming research. As a result, people often rely on multiple data sources such as travel brochures, blogs and vlogs to form a holiday plan and retrieve the top-rated points of interests (POI) of a site. However, these mediums do not hold the resources to provide POIs tailored according to the traveller's preferences ,and constraints and a tourist has to compile a timetable independently [5].

In literature, offering tourists a personalised route composed of POIs has been defined as the tourist trip design problem (TTDP). The TTDP comprises ranking and selecting POIs that might interest the user and create a feasible plan. This is an NP-hard problem where complete algorithms only manage to optimise with a small number of POIs. Therefore, many approximate algorithms, namely heuristic and meta-heuristic approaches, work to converge solutions with complex alternatives to this problem.

Nevertheless, the few existing systems that provide users with an itinerary or activity plan require a lengthy process of manually gathering the users' likes and constraints or information from past trips.

Therefore, we will try to answer the following research question:

Can a system automatically recognise a tourist's travel preferences and use this information to generate a personalised itinerary for a holiday?

1.1 Aims and Objectives

This dissertation aims to build an application that generates a personalised holiday plan according to the user's travel dates and constraints. The following are the objectives:

- **Objective 1 (O1)**: Investigate techniques to build travel interest profiles automatically from social media interactions.
- **Objective 2 (O2)**: Explore different optimisation algorithms for building personalised travel itineraries using the generated travel interest profiles.
- **Objective 3 (O3)**: Evaluate the performance of the personalised travel itinerary generator with real users through in-depth semi-structured interviews.

We will conduct the interviews by generating a personalised and non-personalised timetable for a holiday in Malta to have prior knowledge of the POIs and compare the effects of both itineraries.

2. BACKGROUND AND LITERATURE REVIEW

This section first discusses automatic user profiling to represent travel preferences, then overview and formalise the TTDP research area.

2.1 User Preference Gathering

User profiles are a virtual representation of a user containing their characteristics [4]. In addition, some tourist planners make use of such a technique to personalise the results of their system. For example, Wörndl et al. [26] required the upcoming tourists to input their preferences manually by rating six categories on a scale of 0 to 5: *Sights and Museums, Night Life, Food, Outdoors and Recreation and Shopping.* Including a manual input of user preferences resulted in high user satisfaction since their timetable was very customised.

In 2018, Lim et al.[13] demonstrated how implementing personalisation in their algorithm, PersTours, helped portray real-life scenarios more accurately. The authors built a system where the tourist's level of interest in a specific category is dependant on their time spent at such POIs, relative to the average user. First, they gathered information from the user's past trips from the social media platform Flickr. Then, they evaluated their algorithm using the Root-Mean-Square Error (RMSE), representing the time deviation of past trips and PersTours results from Flickr. Although their results show the PersTours outperforms other applications that use frequency-based user interest, this approach requires users to use Flickr and post information about their past trips on the platform.

Nguyen et al.[16] developed an Android chat application called STSGroup that gathers user's preferences and resolves conflicts between tourists by understanding the messages sent in a group chat. They provided an example of students travelling to South Tyrol (Italy), which gathered information such as the users' mood and recommended POIs from their conversations. Other users in the group chat rate their suggestions through a voting system as the system uses raking and logistics to calculate the ideal group preferences in the background. As a result, 86.7% of the test users showed satisfaction with the suggestions.

The average internet user has gone from being a passive content absorber to a content producer through social media. TTDP solutions can use this advantage and provide a fully automated activity plan based on the user's characteristics. The following are some methods for user profiling and information gathering from the user's social media.

Instagram has a significant effect on the tourism industry. Sharing photos of amazing sights and landscapes influence the way people choose their POIs[21]. Therefore, a system that uses tourist's social media photos could infer the user's preference.

Guntuku et al. [9] performed an analysis on the relationship between a user's characteristics and online images. They found that the media on the social media profile can predict the big five personality traits; conscientiousness, extraversion, neuroticism, agreeableness and openness. The performance graded by the Pearson correlations tests were 0.530 and 0.566 for prognosticating neuroticism and conscientiousness, respectively. Therefore, image classification techniques could provide an automatic preference gathering system.

2.2 Single Route and Multi Route Planners

The Orienteering Problem (OP), introduced by Tsiligirdes [22], is the foundation of single route planners in observance of the sport, orienteering. There are various types of OPs that include different constraints, such as time windows and time dependency [8]. The OP can be represented as a travelling salesman problem with profits.

There are numerous Evolutionary Algorithms (EA) proposed to solve OP. [12, 24]. EAs are algorithms based on natural evolution which use a fitness score to get to the best solution of a problem, in this case, the TTDP [8].

Particle Swarm Optimisation-based (PSO) systems provide prevalent OP solutions with fast computing time [28]. Sevkli et al. [30, 18] tested out two PSO variants: Strengthened Particle Swarm Optimization (StPSO) and Discrete Strengthened Particle Swarm Optimization (DStPSO). These two algorithms introduce pioneering particles, which first perform a local search-based technique called Reduce Variable Neighborhood Search (RVNS) between all the particles and then assign a random velocity. These PSO algorithms obtains either the best or competitive solutions compared with other algorithms such as Ant Colony and Genetic Algorithms when tested on the Tsiligirides [22, 3] dataset of predefined nodes.

A novel approach in 2018 by Kobeaga et al. [12] was able to achieve competitive solutions for medium-sized instances of over 400 nodes and find new best-known solutions for large datasets using the steady-state genetic algorithm. The algorithm also implements a local search, which aims to reduce travel time. Santini et al. [17] introduced a heuristic algorithm based on adaptive extensive neighbourhood search. When they evaluated their system, results showed that EA solutions such as the Kobeaga's GA [12] found slightly more suitable solutions, while their algorithm had a lower average gap between many solutions.

In real-life scenarios, POIs have time constraints that allow them to be visited only during specific hours, such as opening and closing hours or public holiday constraints. Traditional OP is not able to cater for such problems. A single route variant of the OP which solves these issues is the Orienteering Problem with Time Windows (OPTW) [7].

Kantor et al. [11] provided the first attempt towards the OPTW [23]. They developed two algorithms; Insertion and depth-first search. The former algorithm solves the path by selecting a POI with the highest score over-insertion cost incrementally. On the other hand, the depth-first search algorithm gathers parallel tree-based solutions simultaneously and iteratively adds new POIs as long as they follow a set of constraints. Their evaluation showed significant improvements of the second algorithm over the insertion.

When travelling between two POIs, the travel time may depend on certain variable time constraints such as the traffic levels and waiting time [10]. The Time-Dependent Orienteering Problem (TDOP) introduced by Fomin et al. [6] is the single route variant of OP, which considers these scenarios since traditional OPand OPTW does not [8]. In 2011, Abbaspour et al. [1]provide a solution for the Time-Dependent Orienteering Problem with Time Windows, which combines the two previously mentioned OP variants (TDOPTW). They propose two adaptive genetic algorithms and multi-modal shortest pathfinding evaluated in the city of Tehran.

The solutions available from what we discussed in the previous sections can only generate a single efficient path for a tourist's holiday. The Team Orienteering Problem (TOP) [2] is a variant of the OP, which allows for solving the TTDP with multiple days [20]. The system generates a full itinerary for the tourist, with a maximum total score of all routes [10].

Several solutions use PSO-based algorithms to solve the TOP. Muthuswamy et al. [15] developed a discrete version of the PSO (DPSO) which can generate n routes where $2 \geq n \geq 4$. The algorithm consists of two procedures; Random initialisation of n-1 routes. The n^{th} route is based on partial randomness and the current score divided by the current distance of each particle after updating the velocity. The particles use RVNS and 2-opt techniques to communicate with each other as local search techniques. The authors evaluated their work by comparing the algorithm to seven TOP heuristics in which DPSO performed competi-

tively across all applied benchmark data sets [7].

A few years later, Dang et al. wrote another PSO inspired algorithm (PSOiA) for the TOP. They evaluated their work using an interval graph model, which showed how to examine a more extensive search space faster [8].

Besides swarm-based algorithms, an algorithm by Sylejmani et al. [19] used the trajectory-based tabu search to solve a Multi Constrained Team OPTW. Their system followed three steps in order to generate an activity plan: a new activity is added as a node to the trip using *Insert*, a node is exchanged with a new activity using *Replace* and two nodes swap with each other using *Swap*.

2.3 Conclusion

We have concluded that it is possible to gather characteristics from social media throughout this research. Therefore, we will introduce this technology to generate user profiles as part of a constraint with the objective function of the optimisation algorithm.

Since the evolutionary algorithms, PSO and GA, resulted in competing solutions on the Tsilgidres Dataset and are used in novel solutions [28, 25], we will compare both algorithms to see which one to use as the baseline for our TDOPTW application.

3. METHODOLOGY

This section will elaborate user-profiling methods, the itinerary generator and the implementation used to build this application. Figure 1 outlines the overall process of our personalised itinerary generation framework.

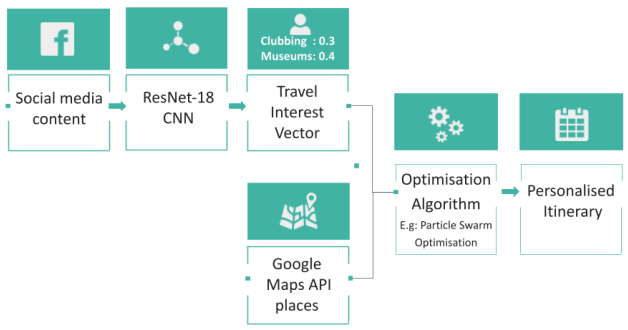

Figure 1: Personalised itinerary generator

3.1 Generating the User Profile

Many people are present on social media posting pictures of their travel moments [14]. For this reason, we opted to build a user profile from the user's online presence. A travel interest vector is made up a vector of six integer values

$$< v_0, v_1, v_2, v_3, v_4, v_5 >$$

representing the user profile. Each element represents the following POI categories 0 Beach,1 Nature,2 Shopping,3 Museums,4 Clubbing and 5 Bars.

At the start of the application, the travel interest vector is initialised with zero values, and the app increments a category whenever the user's content matches. v_0 to v_3 represents morning categories and v_4, v_5 represent evening categories. After the data collection process is complete, the morning and evening vector values are normalised independently.

Facebook's API that allows users to connect both their Facebook and Instagram accounts and request content from the user with their permission. The app requests the photos and the liked pages used to populate the user's travel interest vector.

Transforming the liked pages into the travel interest vector: The API's documentation contains a whole list of possible page categories. The app iterates through all of these user's liked page categories and increments a value in the travel interest vector whenever the Facebook result matches. For example, if a user likes a page with class 'DJ', the user's clubbing vector value is incremented, and if a page is labelled as a 'Mountain', the app increments the user's nature vector value.

Transforming the user's photos into the travel interest vector Convolutional Neural Networks have become a standard for classifying an image because of their high accuracy [29]. Therefore, we decided to test out two approaches for classifying the photos into the app's six categories.

Zhou et al. [29] trained several CNNs for scene recognition and generic deep scene features for visual identification. However, the places365 models are not explicitly trained on the six categories of our application. Therefore, we need to carefully map the 365 categories with our six application's categories. That is why we introduced a Tensorflow Keras sequential model, explicitly trained on the six application's categories to compare.

The pretrained places365 models are pretrained models trained on the places365-standard dataset of about 1.8 million images to classify an image into 365 different scene categories. We used the Resnet places365 models, Resnet-18 and Resnet-50 since they achieved the highest top-5 validation accuracy on the places365 dataset. The Resnet 18 comprises 18, and the Resnet 50 comprises 50 convolutional layers. They both converge to an output layer representing the 365 output categories.

The Tensorflow Keras sequential model contains three convolutional layers with a rectified linear unit (ReLu) activation function. A pooling layer follows each to lower the input volume's spatial dimension for the upcoming layers. The first layer is a rescaling layer that resizes an image to 180×180 pixels. The final layer represents a flattening layer and two dense layers to reduce the outputs to the six application categories, and another representing the 'None' classification.

The dataset contains 3600 public internet images representing the seven classes: Beach, Nature, Museums, Shopping, Clubbing and Bars and None. The tensor library provides tools to Split the dataset into a training and validation set Distribute the photos into batches of 32 Cache the dataset to memory to prevent I/O blocking All of the images were resized to 180x180 pixels, and the RGB values were normalised from zero to one. Since the dataset is small compared to the places 365 models, the training process is prone to overfitting. Data augmentation generates additional samples using random transformations on the dataset. We also added a dropout layer to the model randomly drops sets the input values of the neuron. These two techniques help the model avoid overfitting.

3.2 Producing the activity plan

The problem of our itinerary planner algorithm is mathematically formulated as follows. A tourist trip is made up of some pre-defined user constants alongside the travel interest vector. The predefined constants are:

M:	The number of travelling days.
C:	The activity pace.

The objective function of our itinerary planner is:

$$\text{MAX} \sum_{m=0}^{M} (S_{D_m} + S_{E_m})$$

where:

m	Travelling day (m=1,2,..., M)
D_m	Morning section of day number m
E_m	Evening section of day number m
S_{D_m}	Score of the morning section D_m
S_{E_m}	Score of the evening section E_m

A day is made up of the morning D_m section and the evening E_m section. The timetable suggests a POI in the morning, then somewhere to eat, and the rest is dependant on the activity pace C. That is why the morning section is made up of $C + 2$ tourist attractions. The evening section suggests a place to eat and a POI; therefore, the evening section is just made up of 2.

$$D_m = Y_i + Y_f + C(Y_i) \text{ and } E_m = Y_f + Y_j$$

i	Morning attraction (i = 1,..., n_1)
j	Evening attraction (j = 1,..., n_2)
f	Food Place (f = 1,..., n_3)
$Y_{i\|f\|j}$:	Number of times POI is visited.

Constraints

$\sum_{m=0}^{M} \sum_{i=0}^{n_1} Y_i \leq 1$	Ensures morning POIs not visited more than once.
$\sum_{m=0}^{M} \sum_{j=0}^{n_1} Y_j \leq 1$	Ensures evening POIs not visited more than once.

The score S_{D_m} or S_{E_m} is calculated using

$$S_{D_m|E_m} = \frac{1}{T} + R + V$$

where:

T	Distance between POIs of day m
R	Average rating of POIs of day m
V	How much POIs match with the user's travel interest vector.

3.2.1 Optimisation Algorithms

PSO and GAs are two meta-heuristics that use a population to converge to a fit solution. Therefore, they require an initial random generation of possible timetables. In our algorithm, we introduce a method of randomisation bias. With this technique, the randomness of the initial population is weighted based on the place's rating and the place's number of ratings. This bias gives a head start to the algorithm rather than just starting optimising from purely random itineraries, highly likely to be of bad quality.

Particle Swarm Optimisation.

In PSO, the whole population is referred to as the swarm, whilst a single member a particle. Each particle has a 2-dimensional position(P) vector representing the current timetable solution and a 2-dimensional velocity(V) vector expressing the direction of the particle during its search period.

The algorithm has six integer parameters including the number of particles and the number of iterations. The personal acceleration (PA) affects how far away the particle moves from the personal best position (PB). The global best acceleration (GA) attracts the global best position (GB) of the whole swarm. The inertia (I) constant helps the particle explore new solutions and escape the local minima through randomness.

At each iteration, the new velocity is calculated using

new velocity = I + (PA * (PB - P)) + (GA * (GB - P))

the new position is calculated using

new position = P + new velocity

After a few iterations have passed, particles use their velocity and move towards the optimum position. We demonstrate the framework of our PSO algorithm in algorithm **??**.

Genetic Algorithms.

Genetics algorithms use biological terms to describe their attributes. For example, a timetable solution in population is referred to as a chromosome **??**.

In PSO, the algorithm optimises by allowing each particle to move closer to the global best every iteration. In comparison, in GAs, first, the best chromosomes known as the elites are selected from each iteration. Then, three techniques, namely selection, mutation, and crossover, are applied to generate the next population.

We used the geneticalgorithm2 package [1], which allowed us to use the same score function and the random bias to initialise the particles. The algorithm has 7 parameters. The parents portion represents the number of parents who will reproduce and create the next generation. The mutation probability determines the chance a POI in a chromosome will be replaced by a random value to which the algorithm will converge less quickly and explore more of the search space. The crossover probability will affect the chance that part of its solution goes to the child. Finally, the elite ratio determines how much of the best chromosomes in an iteration make it to the next iteration. There are many types of crossovers techniques. In this algorithm, we explored *one point, two point, uniform and shuffle crossover.* The algorithm produced the following steps:

Step 1: Initialise the first population using random bias.
Step 2: Select the best chromosomes from the population.
Step 3: Select the elite particles that will make it to the next iteration.
Step 3: Apply Crossover, Mutation and Selection on the population.
Step 4: Check if the number of iterations has exceeded.

4. REFERENCES

[1] Abbaspour, R. A., and Samadzadegan, F. Time-dependent personal tour planning and scheduling in metropolises. *Expert Systems with Applications 38*, 10 (sep 2011), 12439–12452.

[1] https://pypi.org/project/geneticalgorithm2/

[2] Chao, I. M., Golden, B. L., and Wasil, E. A. The team orienteering problem. *European Journal of Operational Research 88*, 3 (feb 1996), 464–474.

[3] Chen, Z., Shen, H. T., and Zhou, X. Discovering popular routes from trajectories. In *Proceedings - International Conference on Data Engineering* (2011), pp. 900–911.

[4] Cufoglu, A. User Profiling-A Short Review. Tech. Rep. 3.

[5] De Choudhury, M., Feldman, M., Amer-Yahia, S., Golbandi, N., Lempel, R., and Yu, C. Automatic construction of travel itineraries using social breadcrumbs. In *HT'10 - Proceedings of the 21st ACM Conference on Hypertext and Hypermedia* (2010), pp. 35–44.

[6] Fomin, F. V., and Lingas, A. Approximation algorithms for time-dependent orienteering. *Information Processing Letters 83*, 2 (jul 2002), 57–62.

[7] Gavalas, D., Konstantopoulos, C., Mastakas, K., and Pantziou, G. A survey on algorithmic approaches for solving tourist trip design problems. *Journal of Heuristics 20*, 3 (jun 2014), 291–328.

[8] Gunawan, A., Lau, H. C., and Vansteenwegen, P. Orienteering Problem: A survey of recent variants, solution approaches and applications, dec 2016.

[9] Guntuku, S. C., Lin, W., Carpenter, J., Ng, W. K., Ungar, L. H., and Preotiuc-Pietro, D. Studying personality through the content of posted and liked images on Twitter. In *WebSci 2017 - Proceedings of the 2017 ACM Web Science Conference* (jun 2017), Association for Computing Machinery, Inc, pp. 223–227.

[10] Herzog, D. A. A User-Centered Approach to Solving the Tourist Trip Design Problem for Individuals and Groups. Tech. rep., 2020.

[11] Kantor, M. G., and Rosenwein, M. B. The orienteering problem with time windows. *Journal of the Operational Research Society 43*, 6 (1992), 629–635.

[12] Kobeaga, G., Merino, M., and Lozano, J. A. An efficient evolutionary algorithm for the orienteering problem. *Computers and Operations Research 90* (feb 2018), 42–59.

[13] Lim, K. H., Chan, J., Leckie, C., and Karunasekera, S. Personalized trip recommendation for tourists based on user interests, points of interest visit durations and visit recency. *Knowledge and Information Systems 54*, 2 (feb 2018), 375–406.

[14] Miller, D., Sinanan, J., Wang, X., McDonald, T., Haynes, N., Costa, E., Spyer, J., Venkatraman, S., and Nicolescu, R. *How the World Changed Social Media*. UCL Press, feb 2016.

[15] Muthuswamy, S., and Lam, S. S. Discrete particle swarm optimization for the team orienteering problem. *Memetic Computing 3*, 4 (dec 2011), 287–303.

[16] Nguyen, T. N., and Ricci, F. A chat-based group recommender system for tourism. *Information Technology and Tourism 18*, 1-4 (apr 2018), 5–28.

[17] Santini, A. An adaptive large neighbourhood search algorithm for the orienteering problem. *Expert Systems with Applications 123* (jun 2019), 154–167.

[18] Sevkli, Z., and Sevilgen, F. E. Discrete particle swarm optimization for the orienteering problem. In *2010 IEEE World Congress on Computational Intelligence, WCCI 2010 - 2010 IEEE Congress on Evolutionary Computation, CEC 2010* (jul 2010), IEEE, pp. 1–8.

[19] Sylejmani, K., Dorn, J., and Musliu, N. A Tabu Search approach for Multi Constrained Team Orienteering Problem and its application in touristic trip planning. In *Proceedings of the 2012 12th International Conference on Hybrid Intelligent Systems, HIS 2012* (2012), pp. 300–305.

[20] Sylejmani, K., Dorn, J., and Musliu, N. Planning the trip itinerary for tourist groups. *Information Technology and Tourism 17*, 3 (sep 2017), 275–314.

[21] Terttunen, A. The influence of Instagram on consumers' travel plan-ning and destination choice. Tech. rep., 2017.

[22] Tsiligirides, T. Heuristic methods applied to orienteering. *Journal of the Operational Research Society 35*, 9 (sep 1984), 797–809.

[23] Vansteenwegen, P., Souffriau, W., and Oudheusden, D. V. The orienteering problem: A survey, feb 2011.

[24] Wang, X., Golden, B. L., and Wasil, E. A. Using a genetic algorithm to solve the generalized orienteering problem. *Operations Research/ Computer Science Interfaces Series 43* (2008), 263–273.

[25] Wisittipanich, W., and Boonya, C. Multi-objective Tourist Trip Design Problem in Chiang Mai City. In *IOP Conference Series: Materials Science and Engineering* (jul 2020), vol. 895, Institute of Physics Publishing, p. 012014.

[26] Wörndl, W., Hefele, A., and Herzog, D. Recommending a sequence of interesting places for tourist trips. *Information Technology and Tourism 17*, 1 (mar 2017), 31–54.

[27] WTTC, W. T. Travel & Tourism: Global Economic Impact & Issues 2018, 2018.

[28] Yu, V. F., Redi, P. A. A. N., Jewpanya, P., Gunawan, A., Yu, V. F. ., Redi, P. A. A. N. ., Jewpanya, P. ., Yua, V. F., Perwira Redia, A. A. N., and Jewpanyaa, P. Selective discrete particle swarm optimization for the team Selective discrete particle swarm optimization for the team orienteering problem with time windows and partial scores orienteering problem with time windows and partial scores Citation Citation S. Tech. rep., 2019.

[29] Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., and Torralba, A. Places: A 10 Million Image Database for Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence 40*, 6 (jun 2018), 1452–1464.

[30] Şevkli, A. Z., and Sevilgen, F. E. StPSO: Strengthened particle swarm optimization. *Turkish Journal of Electrical Engineering and Computer Sciences 18*, 6 (nov 2010), 1095–1114.