Adaptive Trip Recommendation System: Balancing Travelers Among POIs with MapReduce

Sara Migliorini, Damiano Carra and Alberto Belussi
Computer Science Department, University of Verona, Verona, Italy
Email:{name.surname}@univr.it

***Abstract*—Travel recommendation systems provide suggestions to the users based on different information, such as user preferences, needs, or constraints. The recommendation may also take into account some characteristics of the *points of interest* (POIs) to be visited, such as the opening hours, or the peak hours. Although a number of studies have been proposed on the topic, most of them tailor the recommendation considering the user viewpoint, without evaluating the impact of the suggestions on the system as a whole. This may lead to oscillatory dynamics, where the choices made by the system generate new peak hours.**

This paper considers the trip planning problem that takes into account the balancing of users among the different POIs. To this aim, we consider the estimate of the level of crowding at POIs, including both the historical data and the effects of the recommendation. We formulate the problem as a multi-objective optimization problem, and we design a recommendation engine that explores the solution space in near real-time, through a distributed version of the Simulated Annealing approach. Through an experimental evaluation on a real dataset, we show that our solution is able to provide high quality recommendations, yet maintaining that the attractions are not overcrowded.

I. Introduction

Traveling is part of many people leisure activities, and an increasing fraction of the economy comes from the tourism. Vising a city is a common choice for a short-term trip: besides the well known destinations, such as New York or Paris, many cities are becoming popular destinations, for instance, during the weekend, or as an intermediate stop while reaching other places. Each destination contains many attractions, or *Points of Interests* (POIs), which are listed in different sources. For instance, travel guides, such as Lonely Planet, provide different suggestions based on the available time. Other options are the Location Based Social Networks (LBSNs) [1], which collect the travellers' experiences to derive popular attractions. Still, once the tourists have a list of POIs to visit, how can they make the most of them given their available time?

Trip recommendation systems deal with this kind of issues. In their essence, these systems need to solve an optimization problem [2], such as the Traveling Salesman Problem, which is NP-hard. Very often, the trip recommendation systems try to provide a solution taking into account not only the tourist available time, but also other elements, such as their personal interests, or budget [3]. Therefore, these systems need to solve a multi-objective optimization problem, whose complexity is further increased. The solutions proposed in literature usually deal with well-known heuristics for local optimization: they translate the user requirements to an utility function, and they adopt different techniques (e.g., gradient descent) to explore the solution space. Recent works focused on more complex scenarios that take into accounts the user needs and constraints: for instance, the POI opening hours may determine its position in the sequence of POIs to be visited [4]. The aim of the trip recommendation system becomes to *tailor* the suggestions to the specific user.

Limitation of the prior work. The proposed solutions concentrate their attention to the user needs and viewpoints: the systems take as input the user preferences and some information about the POIs, and provide a recommended trip. The information about the POIs are "static", such as the opening hours, or an estimate of the busy periods (see Sect. II). The fact that the suggestions have an impact on the status of the POIs is not considered in the recommendation engine. In other words, the optimization is based on the users, not on the system as a whole. For instance, if the recommendation system considers the busy hours of the POIs of the previous day or week, it will generate trips trying to avoid them. This may generate other busy hours, since many of the users may be directed to a specific POI at the same time. This oscillatory dynamics have been observed in routing algorithms that take into accounts the current state of the routes [5]. In order to avoid such dynamics, the system should estimate the *effect of the trip recommendation on the system itself.*

Proposed approach. In this paper we consider the trip planning problem that takes into account, besides the user preferences and the system constraints, the balancing of users among the different POIs. The recommendation engine needs to consider the prediction of the user presence at the POIs. The quality of the prediction determines the quality of the recommendation: the prediction is based on historical data, as well as the recommendations made so far by the system itself. There are a number of challenges that need to be faced to design such a system. First, the user requests are usually issued by a mobile application, where the user expects a near real-time response: the solution space, therefore, should be explored in a limited timeframe. Another issue regards the necessity to understand the impact of the estimation error – due to some unpredictable user behavior – on the balancing process. Finally, in order to increase the effectiveness of the recommendations, the constraints used by the system for comparing possible solution instances should include spatial properties, like for example the total trip distance computed on a network with different traveling modes.

Key contributions. The contributions of our work are the following: (1) we formulate the online optimization problem, where we consider the current estimation of the user visiting the different POIs as part of the input of the recommendation system. (2) We design and implement an efficient solution engine that works in near real-time. The solution is based on a parallel version of the Simulated Annealing approach, using the MapReduce programming framework. (3) We evaluate the trip recommendation system with a dataset collected from the tourist information office of the city of Verona. The dataset contains the visits to the POIs included in a set of city passes.

II. Related Work

This section reviews the related works focusing on two main topics: (i) trip recommendation, and (ii) computational aspects of the solution of optimization problems.

Recommendation systems. This topic has received a lot of attention in recent years, therefore the related literature is vast. Here, due to space constraints, we highlight some representative works based on the taxonomy provided in two recent surveys [1] [6]. The interested reader can find more details in such surveys and the references therein.

The main problem to consider is the identification of the POIs and their relevance. The data used to find POIs can be gathered from different sources, such as user check-in behaviours [7], [8], crowdsourced digital footprints [9], [10], GPS data [11], [12], or it can be inferred by using geographical or social correlations of visited POIs [13], [14]. Once the system has the list of POIs, it needs to select the subset of POIs that are relevant to the user. The recommendation may take into account multiple constraints [2], [15] or constraints related to time [4], [16]. The POIs can be used to build semantically enriched trajectories – for a survey on the topic, please refer to [17]. All the above systems are focused on the user viewpoint to provide a tailored recommendation. Only some of them include geographical consideration in building the itinerary, and none of them adapts the solution considering the number of users that can be present at the POIs.

The current crowding of the POIS is only considered in [3]. Nevertheless, the proposed system bases its recommendations on instantaneous information, therefore it may generate new peak hours at the different POIs. Moreover, the authors do not consider the geographical aspects in building the itinerary. To the best of our knowledge, our work is the first that takes into consideration the impact of the recommendations on the current and future level of crowding, so that to balance the users among the POIs.

Optimization problem. Approximate solutions of optimization problems have been extensively treated in the literature. Hoos et al. [18] provide a broad view of the techniques and the solutions adopted so far.

Since we are interested in a near real-time system, we focus on some works that deal with the parallel implementation of a specific technique, i.e., the Simulated Annealing (SA). Frequently the Asynchronous Approach [19], [20] is adopted, where different workers executes independent SA using different starting solutions, and the best solution among them is reported. Inspired by such an approach, the authors in [21], [22] propose different MapReduce implementations, where the computations is divided among MAP and REDUCE tasks in different ways. The solution of multi-objective optimization problems using SA have been considered in [23], [24], and its parallel implementation in [25]. To the best of our knowledge, these parallel implementations have never been adapted to the MapReduce framework. We take inspirations from the above mentioned works to design a MapReduce implementation of a multi-objective optimization problem.

III. Problem formulation

In this section we provide the necessary definitions and formalize the trip planning problem we consider.

Definition 1 (Point of interest)**.** *A point of interest (POI) p represents an attraction reachable by users. It is characterized by several attributes, such as the admission fee, or the opening hours. Among these, we consider: the spatial coordinates defining its position on the Earth surface, which we denote with p^c, and the* duration of a visit*, denoted by $p^v(t)$, which depends on the time t when the visit starts.*

The dependency on t is necessary since $p^v(t)$ is influenced by many factors, such as the day of the week, and the number of people currently visiting p. We will show in Sect. IV how we compute (and update) the value of $p^v(t)$. For the purposes of this paper, the set of POIs $\mathcal{P}$ that can be considered for building a trip is assumed to be known and fixed.

Definition 2 (Trip)**.** *A trip τ is an ordered collection of POIs, i.e., $\tau = \langle p_1, p_2, \dots, p_n \rangle$, where n indicates the number of POIs contained in τ, $|\tau| = n$.*

The set $\mathcal{T}$ of all possible trips contains all the possible ordered combination of POIs in $\mathcal{P}$, for any cardinality of τ.

Definition 3 (Path)**.** *Given any two spatial coordinates c_i and c_j, and a travel mode m (e.g, walking, public transportation), a path $\pi(c_i, c_j, m)$ is a continuous portion of a transport network that connects the points whose location is defined by c_i and c_j. The path is characterized by the travel distance, $\pi_{td}(c_i, c_j, m)$, and by the travel time, $\pi_{tt}(c_i, c_j, m)$.*

Note that, in order to maintain the notation simple, we may not indicate the dependency of π_{td} (π_{tt}) on the travel mode, which it is specified by the user when she submits the query.

Definition 4 (Recommendation query)**.** *Users looking for a recommendation submit a query $\mathcal{Q}$ to the system by specifying the following constraints:*

- *the initial coordinates c_0 where the trip begins;*
- *the time t_0 at which the trip will start;*
- *the maximum trip duration $TD_{\max}$;*
- *the travel mode m.*

In order to reply to such a query, the system needs to compute a set of values that drives the trip selection. We start considering the main constraint, i.e., the total time of the duration of the trip should be less than $\mathrm{TD}_{\max}$. To this aim, we introduce a fictional POI p_0, which corresponds to the user initial position, and we set $p_0^c = c_0$ and $p_0^v(t_0) = 0$. We denote with t_i the time of arrival at p_i, the i-th POI of the trip, which can be computed considering the time t_{i-1}, the visit time of the previous POI and the travel time between the two POIs, i.e., $t_i = t_{i-1} + p_{i-1}^v(t_{i-1}) + \pi_{tt}(p_{i-1}^c, p_i^c),\ i \geq 1$. Note that $t_1 = t_0 + p_0^v(t_0) + \pi_{tt}(p_0^c, p_1^c) = t_0 + \pi_{tt}(c_0, p_1^c)$, which represents the staring time of the trip plus the travel time between the user position and the first POI. We can now define the total trip time λ_τ for a trip τ as:

$$\lambda_\tau(c_0, t_0) = \sum_{i=1}^{|\tau|} \left(\pi_{tt}(p_{i-1}^c, p_i^c) + p_i^v(t_i) \right), \tag{1}$$

When exploring the solution space, the system will consider the trips for which $\lambda_\tau(c_0, t_0) < \mathrm{TD}_{\max}$. The exploration is guided by the values of the objective function. In the following, we consider a set of possible optimization criteria that can be minimized. For simplicity, we focus on three criteria: adding more objective functions is cumbersome.

Definition 5 (Objective functions)**.** *Given a trip τ, the objective functions f_n, f_{tt} and f_{td} denote the number of locations* not *visited during the trip, the estimated trip travel time, and the total distance travelled during the trip, respectively. They are computed as:*

$$\begin{aligned} f_n &= |\mathcal{P}| - |\tau|, \qquad f_{tt} = \sum_{i=1}^{|\tau|} \pi_{tt}(p_{i-1}^c, p_i^c), \\ f_{td} &= \sum_{i=1}^{|\tau|} \pi_{td}(p_{i-1}^c, p_i^c). \end{aligned} \tag{2}$$

We are now ready to define the trip planning problem, which can be cast as an optimization problem:

$$\begin{aligned} &\underset{\tau}{\text{Minimize}} && \langle f_n, f_{tt}, f_{td} \rangle, \\ &\text{subject to} && \lambda_\tau(c_0, t_0) < \mathrm{TD}_{\max} \end{aligned} \tag{3}$$

Note that the global objective function we would like to minimize is a composition of objective functions, and it can be defined as $\bar{f} : \mathcal{T} \to \mathbb{R}^3$. We are therefore in the context of *multi-objective* optimization, in which is not possible to define a total order. We need to introduce a *dominance* relation to partially order the set of possible solutions. A trip τ_i dominates a trip τ_j, denoted $\tau_i \prec \tau_j$, if at least one of the composing objective functions is smaller for τ_i than for τ_j, while the others are equivalent. The results of the optimization problem will be the set of mutually non-dominating trips, i.e., $res(\mathcal{Q}) = \{\tau \in \mathcal{T} \mid \nexists \tau_0 \in \mathcal{T} \text{such that } \tau_0 \prec \tau\}$.

Considering the cardinality of the set containing all the possible trips, $\mathcal{T}$, the solution space to explore to provide a recommendation is very large. In addition, note that the total trip time depends on the POI visit duration, which depends on the time when the visit starts, thus the solution space further increases. For this reason, our search is based on heuristics.

IV. Proposed system

Our proposed recommendation system (Fig. 1) has two main components: an offline analysis of the user presence in the different POIs, and a recommendation engine based on a parallel implementation divided into two main stages.

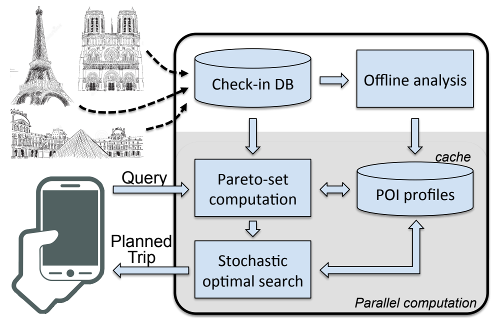

Fig. 1. System architecure.

Offline analysis. The offline analysis processes the records of the visits at the POIs. In particular, it considers the users with identifiers, i.e., the ones that use, for instance, a bundle offer. For these users, we can reconstruct the set of POIs they visited. We can also derive the set of popular trips that can be used by the recommendation engine as a starting point in the search of the optimal solution when replying to a query. The popular trips are stored back in the database, and accessed by the recommendation engine when it processes a query.

From the data, it is also possible to build a set of characterizing measures, such as (i) the average number of visitors inside the POI at different times, and (ii) the average visiting time given a number of visitors inside the POI. These measures are used when we compute adaptively the recommendation.

Exploration of the solution space. The exact solution of (3) is computationally expensive, thus we resort to heuristics. Our solution builds trip recommendations using a dominance-based Multi-Objective Simulated Annealing (MOSA) [24] technique.

With multi-objective optimization, we can define a partial order on the solution based on the concept of dominance. The set of mutually non-dominating solutions is called *Pareto-set*. A solution not dominated by any other solution is called *Pareto-optimum*. The aim of our system is to (i) find the initial Pareto-set given the user query, and (ii) explore the solution space to look for the Pareto-optimum. The exploration is based on the comparison between the current solution with a new potential solution, obtained through a *perturbation*. The perturbation could be, for instance, a POI removal or addition, or a change in the order of the POIs.

The two steps (identification of the starting Pareto-set and exploration) can be done in parallel with the MapReduce framework. We use different mappers for executing independent iterations of the Pareto-set computation, starting from different solutions, and we then use the reducer to compute the final result. Due to space constraints, we leave out the details of the algorithms: the interested reader may find additional information in [26].

TABLE I

STATISTICS ABOUT THE COLLECTED TRIPS. THE COLUMNS REPORT: THE NUMBER OF VISITED POIS, THE NUMBER OF TRIPS WITH SUCH NUMBER OF POIS, THE AVERAGE DURATION OF THE TRIP (HOUR:MIN), THE AVERAGE TRAVEL TIME, AND THE AVERAGE TRAVEL DISTANCE.

$\|\tau\|$	# trips	λ_τ	avg trav. time	avg trav dist.
2	14,520	04:10	00:10	750m
3	31,455	04:20	00:17	1,5Km
4	40,878	06:00	00:26	2,0Km
5	37,900	07:50	00:34	2,7Km
6	28,261	09:00	00:42	3,4Km
7	16,139	10:30	00:51	4,0Km
8	7,823	11:30	00:60	4,7Km
9	3,060	12:00	01:10	5,5Km

Dynamic adaptation. During the computation of the solution for the user, the recommendation system needs to take into account the information about the visiting time – since the solution must be feasible, i.e., its duration must be less than the maximum trip duration $\mathrm{TD}_{\max}$. The visiting time in turn depends on the POI occupancy. With the offline analysis, we have the averages of these values. To adapt to the current occupancy, as the system issues recommendations, it records the choices of the users, and it updates in real-time the estimation of the POI occupancy, and the expected visiting time of each POI, so that the following recomandations will automatically consider this information while exploring the solution space.

V. Case Study and Experiments

We evaluated the proposed systems using real-world traces collected for registered tourists visiting the city of Verona (IT).

Available dataset. The tourist office of the city of Verona offers a *sightseeing city pass* called "VeronaCard": for a given fee, the tourist may visit up to 22 POIs around the city within a specific time-frame (e.g., 24 hours, or 48 hours). Every time a tourist with the pass enters in a POI, a record is created: it contains the VeronaCard number (unique identifier), the timestamp of the entrance and the POI identifier. The dataset includes approximately 1,200,000 records that spans 5 years.

From this dataset, we derive a set of data and measurements that we use in our experiments. We build the trips followed by the tourists with a VeronaCard, i.e., the sequence of visited POIs, obtaining approximately 250,000 trips. Table I shows some statistics related to trips. We grouped trips with the same number of visited POIs, $|\tau|$: for each group, we show its cardinality, the average duration of its trips (considering the first POI as the starting POI), the average travel time and travel distance – we first sum the travel times and travel distances of the paths for each trip, and then compute the average.

We also compute the visiting time for each POI, at different times of the day, and the number of tourists inside each POI. From these values, we compute the averages used in the recommendation system, i.e., we obtain the average number of visitors inside the POI at different times, and the average visiting time given a number of visitors. Fig. 2 shows sample averages for two POIs called "Casa di Giulietta" and "Castelvecchio". As for the average time occupancy (Fig. 2, left), we show the curves for July's Sundays (the average number of visitors computed considering the Sundays in July). As expected, there are two peak hours, in the morning and the afternoon. Interestingly, the peak hours for the two POIs in the afternoon are slightly different.

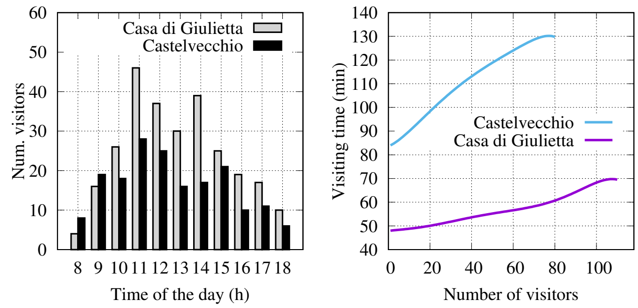

Fig. 2. Left: Average time occupancy for two POIs in Verona (the averages have been computed considering the Sundays in July). Right: Average visiting time for the same POIs (the averages are computed considering the whole dataset).

Similarly, we notice an increasing visiting time as the number of visitor increases (Fig. 2, right), which indicates the impact of crowding in the visiting time.

Experimental methodology. In order to test our system, we need to provide a set of queries. To this aim, we consider our dataset and the trips built from it. For a given day, we consider the trips collected that day: for each trip, we create a query where (i) the initial coordinates at which the trip starts are the coordinates of the first POI, (ii) the time at which the trip begins is the time of the access to the first POI, and (iii) the maximum trip duration is given by the computed trip duration augmented with an estimate of the duration of the last visit.

For that set of queries, we observe the output from two possible perspectives: the POI viewpoint and the trip viewpoint. From the POI viewpoint, we record for each POI the number of visitors over time, and we build the time occupancy curve for that day. From the trip viewpoint, we record the values of the objective functions defined in Eq. (2).

We consider two possible approaches and compare them with the starting scenario *without recommendation*. (i) *Recommendation based only on historical data* (static approach): we use the average occupancy of POIs collected in the previous observation interval (e.g., last year); (ii) *Adaptive Recommendation* (dynamic approach): we use the current estimated POI occupancies, which are updated after every recommendation.

The MapReduce algorithm has been implemented using SpatialHadoop [27], which has been successfully applied to efficiently perform spatial analysis and validation of huge amount of geographical data [28].

POI viewpoint. Fig. 3 shows the number of visitors over time for the POI called "Casa di Giulietta" on February 14th, 2015, with or without a recommendation system. It is interesting to note that a static recommendation simply changes the peak hour with respect to a system with no recommendation, since it uses the average peak hour of the past days, but it does not adapt to the estimated number of users in the POI. Instead, our dynamic recommendation spread the tourists over time.

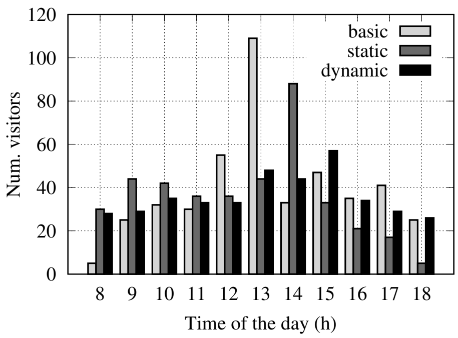

Fig. 3. Number of visits at "Casa di Giulietta" both considering the behavior of the tourists without recommendation (basic) and with the two approaches based on average (static) and adaptive (dynamic) profiles for POIs.

Trip viewpoint. Due to space constraints, we refer the interested reader to [26] for the results.

VI. Conclusion and Future Work

Personalized trip recommendation systems tailor the suggestions to the users based on their constraints and requirements. Nevertheless, they do not consider the impact of the recommendations on the whole system. In this paper we took a step to fill this gap. In particular, we proposed a system that efficiently searches the solution space through a MapReduce implementation of the multi-objects optimization problem and balances the users among different POIs by including the predicted level of crowding. We evaluate our implementation using a real dataset, showing consistent improvements over the paths usually followed by the tourists.

Our road-map includes the evaluation of the impact that errors may have on the predictions of the level of crowding, and the corresponding quality of the recommendations. Moreover, we will consider problems related to the management of different temporal granularities [29] in the computation of the visiting time of each POI given the current level of crowding. Similarly, as regards to the collection of spatial aspects about previous trips, we will consider issues related to the integration of data with different positional accurancies [30].

Acknowlegments

We will thank the UNESCO office of Verona for providing the datasets of the VeronaCard city pass. This work was partially supported by the Italian National Group for Scientific Computation (GNCS-INDAM). This work has been supported by "Progetto di Eccellenza" of the Computer Science Dept., University of Verona, Italy.

References

[1] Y. Yu and X. Chen, "A survey of point-of-interest recommendation in location-based social networks," in *Workshops at the 29th AAAI Conference on Artificial Intelligence*, 2015.

[2] E.-C. Lu, C.-Y. Chen, and V. Tseng, "Personalized trip recommendation with multiple constraints by mining user check-in behaviors," in *Proc. of ACM SIGSPATIAL*, 2012, pp. 209–218.

[3] X. Wang et al., "Improving personalized trip recommendation by avoiding crowds," in *Proc. of ACM CIKM*, 2016, pp. 25–34.

[4] Q. Yuan et al., "Time-aware point-of-interest recommendation," in *Proc. of ACM SIGIR*, 2013, pp. 363–372.

[5] K. Varadhan et al., "Persistent route oscillations in inter-domain routing," *Computer networks*, vol. 32, no. 1, pp. 1–16, 2000.

[6] S. Zhao, I. King, and M. Lyu, "A survey of point-of-interest recommendation in location-based social networks," *arXiv preprint arXiv:1607.00647*, 2016.

[7] E. Cho, S. Myers, and J. Leskovec, "Friendship and mobility: user movement in location-based social networks," in *Proc. of ACM SIGKDD*, 2011, pp. 1082–1090.

[8] H. Gao, J. Tang, X. Hu, and H. Liu, "Exploring temporal effects for location recommendation on location-based social networks," in *Proc. of the 7th ACM Conf. on Recommender Systems*, 2013, pp. 93–100.

[9] C. Chen et al., "Tripplanner: Personalized trip planning leveraging heterogeneous crowdsourced digital footprints," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 3, pp. 1259–1273, 2015.

[10] H. Gao et al., "Content-aware point of interest recommendation on location-based social networks," in *AAAI*, 2015, pp. 1721–1727.

[11] Y. Zheng and X. Xie, "Learning travel recommendations from user-generated gps traces," *ACM Trans. on Intelligent Systems and Technology*, vol. 2, no. 1, p. 2, 2011.

[12] H. Yoon et al., "Smart itinerary recommendation based on user-generated gps trajectories," in *Proc. of UIC*, 2010, pp. 19–34.

[13] Y. Liu, W. Wei, A. Sun, and C. Miao, "Exploiting geographical neighborhood characteristics for location recommendation," in *Proc. of ACM CIKM*, 2014, pp. 739–748.

[14] M. Ye et al., "Location recommendation for location-based social networks," in *Proc. of ACM SIGSPATIAL*, 2010, pp. 458–461.

[15] G. Adomavicius and Y. Kwon, "Multi-criteria recommender systems," in *Recommender Systems Handbook*. Springer, 2015, pp. 847–880.

[16] J.-D. Zhang and C.-Y. Chow, "TICRec: A probabilistic framework to utilize temporal influence correlations for time-aware location recommendations," *IEEE Trans. on Services Computing*, vol. 9, no. 4, pp. 633–646, 2016.

[17] C. Parent et al., "Semantic trajectories modeling and analysis," *ACM Computing Surveys (CSUR)*, vol. 45, no. 4, p. 42, 2013.

[18] H. Hoos and T. Stützle, *Stochastic local search: Foundations and applications*. Elsevier, 2004.

[19] E. Onbaşoğlu et al., "Parallel simulated annealing algorithms in global optimization," *Jrnl of Global Optimiz.*, vol. 19, no. 1, pp. 27–50, 2001.

[20] Z. Czech and P. Czarnas, "Parallel simulated annealing for the vehicle routing problem with time windows," in *Proc. of Euromicro Workshop on PDP*, 2002, pp. 376–383.

[21] A. Radenski, "Distributed simulated annealing with mapreduce," in *Proc. of EVOSTAR*. Springer, 2012, pp. 466–476.

[22] H. Li and C. Liu, "Prediction of protein structures using a map-reduce Hadoop framework based simulated annealing algorithm," in *2013 IEEE Intern. Conf. on Bioinformatics and Biomedicine*, 2013, pp. 6–10.

[23] B. Suman and P. Kumar, "A survey of simulated annealing as a tool for single and multiobjective optimization," *Journal of the operational research society*, vol. 57, no. 10, pp. 1143–1160, 2006.

[24] B. Suman, "Study of simulated annealing based algorithms for multiobjective optimization of a constrained problem," *Computers & Chemical Engineering*, vol. 28, no. 9, pp. 1849–1871, 2004.

[25] P. McMullen et al., "Using simulated annealing to solve a multiobjective assembly line balancing problem with parallel workstations," *Intern. Jrnl of Production Research*, vol. 36, no. 10, pp. 2717–2741, 1998.

[26] A. Belussi, D. Carra, and S. Migliorini, "Adaptive trip recommandation system," Dept. of Computer Science, University of Verona, Tech. Rep. RR 107/2018, 2018. [Online]. Available: https://iris.univr.it/retrieve/handle/11562/979749/102409/rr_107_2018.pdf

[27] A. Eldawy, "SpatialHadoop: Towards Flexible and Scalable Spatial Processing Using Mapreduce," in *Proc. of the 2014 SIGMOD PhD Symposium*, 2014, pp. 46–50.

[28] S. Migliorini, A. Belussi, M. Negri, and G. Pelagatti, "Towards massive spatial data validation with SpatialHadoop," in *5th ACM SIGSPATIAL Int. Workshop on Analytics for Big Geospatial Data*, 2016, pp. 18–27.

[29] A. Belussi, C. Combi, and G. Pozzani, "Towards a formal framework for spatio-temporal granularities," in *15th International Symposium on Temporal Representation and Reasoning*, 2008, pp. 49–53.

[30] A. Belussi and S. Migliorini, "A framework for integrating multi-accuracy spatial data in geographical applications," *GeoInformatica*, vol. 16, no. 3, pp. 523–561, 2012.