Applied Soft Computing 71 (2018) 268–276

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Creating hard-to-solve instances of travelling salesman problem

Miguel Cárdenas-Montes

CIEMAT, Department of Fundamental Research, Avda. Complutense 40, 28040 Madrid, Spain

ARTICLE INFO

Article history:
Received 19 June 2017
Received in revised form 6 July 2018
Accepted 6 July 2018
Available online 10 July 2018

Keywords:
Travelling salesman problem
Difficulty evaluation
Dirichlet tessellation
Delaunay triangulation
Weibull probability distribution
Genetic algorithm

ABSTRACT

Travelling salesman problem is a classical combinatorial optimization problem. Instances of this problem have been used as benchmark for testing the performance of many proposals of discrete optimizer algorithms. However, the hardness or the difficulty degree to solve the instances has not been usually addressed. In the past the evaluation of the difficulty of the instances has required to obtain a high-quality solution, ideally the optimal one. However, this type of strategy burdens the evaluation with large processing times. In this work, diverse indirect measures for evaluating the hardness to solve instances of the travelling salesman problem are proposed. These evaluations are inferred from the spatial attributes of previously evaluated instances, and later correlated with the hardness of the instances. Finally, where a significant correlation is found, a linear model is built and linked to a genetic algorithm. As a consequence of this work, mechanisms for hardening instances of travelling salesman problem are implemented.

1. Introduction

Travelling salesman problem (TSP) is one of the most popular problems in combinatorial optimization. The quest for searching high-quality solutions to these problems has fostered the development of ground-breaking evolutionary algorithms (see Blum [1] for a review). In parallel numerous variants of TSP [3,2] are the result of the search of difficult variants of the problem for stressing more advanced heuristics and metaheuristics. TSP is an NP-hard problem [4], but it does not mean that all the instances have the same difficulty.

In order to fairly evaluate the performance of heuristics and metaheuristics when optimizing TSP instances, hardness ranks about the TSP instances and mechanisms to increment their hardness ought to be generated. With regards to the metrics to evaluate the hardness of a particular instance, the existence of a phase transition in TSP is cited in [5]. This issue arises from the transformation of the TSP into a binary decision problem under the question, *can an algorithm find a solution with a tour length less than l?*. The phase transition easy-hard-easy related with the difficulty to find a new solution to a TSP instance with a shorter tour length is analysed. Authors place this phase transition at $\frac{l}{\sqrt{N \cdot A}} \approx 0.75$, where N is the number of cities, A the area covered by the cities and l the tour length. Thus the parameter $|\frac{l}{\sqrt{N \cdot A}} - 0.75|$ can be used as hardness indicator of the difficulty to solve a TSP instance. Unfortunately this indicator requires to find the optimal or a quasi-optimal solution of the instance.

The search of the optimal solutions of TSP instances is an intensive computational task for very large instances. Therefore, it makes difficult to frequently use it.

Starting from this work, the correlation between this hardness indicator and the statistical distribution of three spatial attributes of TSP instances is studied. They include the distribution of the normalized areas generated from the Dirichlet tessellation, from the Delaunay triangulation, and the distances between the cities of the instance. The generation of this relationship allows evaluating the hardness of a TSP instances without calculating the optimal tour, and therefore processing time can be saved.

In detail the proposed methodology is as follows. Initially the histogram of the normalized areas generated from the application of the Dirichlet tessellation to the TSP instances is fitted to a Weibull probability distribution. From this adjustment, the two parameters of this distribution can be extracted: shape and scale. Then, the correlation between the hardness of the instances and the shape parameter is analysed. Independently, this procedure is also repeated for the normalized areas generated from the Delaunay triangulation, and the distances between the cities of each instance. In the cases where a high correlation is found, it is used for establishing a model between both variables.

Furthermore, this relationship, expressed as a linear equation, can be easily linked to an evolutionary algorithm aimed at generating hard-to-solve instances. In this work, this possibility is tested through a genetic algorithm. From randomly-generated TSP instances of 100 cities in a squared area of length 200 in arbitrary

E-mail address: miguel.cardenas@ciemat.es

https://doi.org/10.1016/j.asoc.2018.07.010
1568-4946/

units, the genetic algorithm makes evolve them. At each generation, the instances are crossed and mutated, and those with larger hardness indicator, lower fitness, are promoted to the next generation. The final result is a set of hard-to-solve TSP instances. Only the best individual of each run is retained for constructing a catalogue of hard-to-solve TSP instances.

As a consequence of this effort, a mechanism for hardening instances of TSP, in a way that randomly-generated instances can evolve until obtaining arbitrary hard instances, is proposed. The motivation behind this work is to be capable of generating very large TSP instances with the hardness required for benchmarking ground-breaking discrete optimizers.

To the author's knowledge, no similar works aimed at increasing the hardness of TSP instances have been presented and analysed.

The rest of the paper is organized as follows: Section 2 summarizes related work and previous efforts done in this area. The theoretical foundations of the travelling salesman problem are outlined in Section 3.1. A study of the hardness of instances of TSP is presented in Section 3.2. The basic concepts of the Weibull probability distribution are described in Section 3.3. The Dirichlet tessellation and the Delaunay triangulation are presented in Section 3.4. The software and hardware used is described in Section 3.5. Results are presented and analysed in Section 4. Finally, Section 5 contains the conclusions of this work.

2. Related work

From the historical point of view, in [6] two algorithms for solving TSP are presented. This work uses as benchmark two TSP instances with 13 cities and 9 cities respectively. Comparing with the current problem sizes, these tiny problem sizes are revealing how the capacity to produce high-quality solutions has evolved, and therefore, the need of a set of hard TSP instances to fairly evaluate the ground-breaking heuristics and metaheuristics. For a more complete review of the problem, the reader is referred to [3,2] for more information.

In [7], the objective is to predict the relationships between the performance of algorithms and the critical features of instances. The paper provides a methodology to discern if the metadata is sufficient for predicting the behaviour of the instances, easy or hard. In this work, the experimental set-up is composed of instances of 100 randomly-generated cities in a squared area of 400 units. The TSP instances are evolved using an evolutionary algorithm, particularly a genetic algorithm: new instances are created by using uniform crossover and mutation. The criterion for classifying as easy or as hard the instances is based on the effort done by two Lin–Kernighan heuristic methods [8] to solve the TSP instance: chained Lin–Kernighan [9] and Lin–Kernighan with cluster compensation [10].

In [11] a major revision and extension of [7] is done by extending the methodology to the most popular combinational problems: assignment problem, knapsack problem, bin-packing problem, graph problem timetabling and constraint satisfaction.

Beyond the TSP, the existence of phase transition for the NP-complete problems, between regions where the problem is generally easy to solve to other regions where the problem is generally much more difficult to solve, is widely presented in [12–15].

In [16] a study about the capacity to predict the hardness of TSP instances, as defined in [5], from attributes arisen from the statistical distribution of the distance between the cities, the areas generated from the Dirichlet tessellation, and the areas from the Delaunay triangulation is undertaken. These attributes include the minimum and maximum sizes, the mean, variance, skewness and kurtosis of the distribution of the normalized areas from the Dirichlet tessellation, and the areas from the Delaunay triangulation. The same information from the Euclidean distance between the cities is also used as attributes for the hardness classification of the instances. Random Forests is used for classifying TSP instances in hard- or easy-to solve through spatial features [17].

In [18], efforts for finding additional mechanisms for evaluating the intrinsic hardness of the TSP instances can be found. In this work, the instances are evaluated only in the nearby of the optimal solution by applying random walk to study the gradient of the fitness landscape around the optimal position.

3. Methodology

3.1. Travelling salesman problem

The TSP can be expressed as shown in Eqs. (1) and (2).

$$x_{ij} = \begin{cases} 1 & \text{the path goes from city } i \text{ to city } j \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $x_{ij} = 1$ if city i is connected with city j, and $x_{ij} = 0$ otherwise.

For $i = 0, \ldots, n$, let u_i be an artificial variable, and finally take c_{ij} to be the distance from city i to city j. Then TSP can be written as shown in Eq. (2).

$$\begin{aligned} &\min \sum_{i=0}^{n} \sum_{j \neq i, j=0}^{n} c_{ij} x_{ij} && \\ &0 \le x_{ij} \le 1 && i, j = 0, \ldots, n \\ &u_i \in \mathbf{Z} && i = 0, \ldots, n \\ &\sum_{i=0, i \neq j}^{n} x_{ij} = 1 && j = 0, \ldots, n \\ &\sum_{j=0, j \neq i}^{n} x_{ij} = 1 && i = 0, \ldots, n \\ &u_i - u_j + n x_{ij} \le n - 1 && 1 \le i \neq j \le n \end{aligned} \tag{2}$$

The purpose of TSP is to find the shortest tour between a set of cities. In the symmetric version of TSP — variant used in this work —, the cost of joining two cities does not depend on the departure city, only on the pair of cities.

The TSP instances used as benchmarks in this work have been extracted from http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/. Among the instances published in this site, those with optimal solution have been selected. Besides, several instances are selected under the criterion of having 100 cities. This allows a study of the difficulty of instances of equal size. Other instances are selected under the criterion of having a number of cities slightly lower or larger than 100, to facilitate the mapping of the difficulty in function of the number of cities. Instances varying more than two orders of magnitude in relation with the smallest one have been skipped. Hence, efforts to evaluate the difficulty relative among instances which encompass from tens to hundreds of cities are done.

The list of TSP instances includes: ulysses16, ulysses22, bays29, att48, eil51, berlin52, st70, eil76, gr96, kroA100, kroC100, kroD100, rd100, eil101, gr120, ch130, ch150, gr202, pa561 and gr666. The figure in the names indicates the number of cities conforming the TSP instance. With this selection of TSP instances, it is expected to cover a wide range of cases, and therefore, different difficulty instances are evaluated.

3.2. Easy-hard-easy phase transition

In [5], for Euclidean TSP instances in the plane the existence of a phase transition — easy-hard-easy — for the TSP decision problem