

A GENETIC ALGORITHM TO DESIGN TOURISTIC ROUTES IN A BIKE SHARING SYSTEM

I. Baffo ^(a), P. Carotenuto ^(b), G. Storchi ^(c)

^{(a),(b)} Consiglio Nazionale delle Ricerche - Istituto per le Applicazioni del Calcolo “M. Picone”, Via dei Taurini 19, Roma - 00185, Italy

^(c) Università di Roma “Sapienza” – Dipartimento di Scienze Statistiche, Piazzale Aldo Moro 51, 00185 - Roma, Italy

^(a) i.baffo@iac.cnr.it, ^(b) p.carotenuto@iac.cnr.it, ^(c) giovanni.storchi@uniroma1.it

ABSTRACT

The aim of this paper is to study a Bike Sharing Touring (BST) applying a mathematical model known in operation research as Orienteering Problem (OP). Several European Cities are developing BST in order to reduce the exhaust emissions and to improve the sustainability in urban areas. The authors offer a Decision Support Tool useful for the tourist and the service's manager to organize the tourists' paths on the basis of tourists' desires, subject to usable time, place of interest position and docking station location. The model analyzed presents two innovative aspects compared to a classic OP. The first one is that the start and the arrival point of routes aren't necessary coinciding and pre-conditioned. The last one is that the knowledge of tourist tours allows to book the visit to a point of interest and doing so to optimize efficiency of the whole system and not only of the single tourist tour.

Keywords: Orienteering Problem, Bike Sharing Tourist, Decision Support System

1. INTRODUCTION

Bike sharing schemes are short term urban bicycle rental schemes that allow to pick up bicycles at any self-serve bicycles station and return them to any other bicycles station. This structure makes bicycle - sharing ideal for point-to-point trips. The bike sharing process involves several units and parties like bikes, docking stations, station services and users. The characteristics of these systems are often in continue evolution and differ country to country. The European's interest in this kind of mobility, with particular focus on big cities, has increased in the last year. For this reason recently many funds and several scientific research are focused on improving the bike sharing service for the users and for the head of management.

In our research we aim to improve bike sharing service, thanks to a Mobile Tourist Guide (MTG) of new generation, able to support the cyclist or tourist to decide which point of interest to visit and in which sequence. This decisional process is known in literature as Tourist Trip Design Problem (TTDP) and very often

is defined as an extension of another problem called Orienteering Problem (OP).

In the OP, several location have an associated score and have to be visited only once to obtain a total trip score. The objective is to obtain a score as high as possible without violating a given time restriction.

A Tourist Trip Design Problem (TTDP) consists into finding the best tour or more tours taking into account a set of N locations to visit with the respective scores. The score is considered as a representation of the user preferences with respect to that particular point to visit. Moreover, this problem is made more complex by the limitation imposed by users in terms of time. Generally the tourist knows in advance the time that can use in a tourist tour. Others elements of complexity to solve this kind of problem are: opening time of Point of Interests (PoIs), weather conditions, uncertainty about the scheduled visit times caused by queues or service blackout, preferences expressed by the users on the route used to reach the PoI.

In order to face some of these complexities, researchers have proposed several approaches. In paragraph 2 a literature review is presented. In the following two paragraphs the Orienteering Problem is detailed and our solution approach based on Genetic Algorithm is proposed. In the last two paragraphs, experimental results and conclusion are given.

2. LITERATURE REVIEW

This work develops an algorithm that permits the creation of touristic routes, in which each location have associated a score and the tour have to be completed in a specific time T_{max} . The described route-planning problem can be considered as an application of the OP.

In the OP, several locations with an associated score have to be visited only once in order to obtain a total trip score. The objective is to obtain a total trip score as high as possible without violating a time restriction.

There has been works on exact methods that have yielded solutions to smaller sized problems. Due to the computational limitations of the exact algorithms, the heuristic procedures were explored.

The first heuristics were proposed by Tsiligirides (1984) and are known as the S-algorithm and the D-algorithm. The S-algorithm uses the Monte Carlo method to construct routes using probabilities correlated to the ratio between node score and node distance from the current node. The D-algorithm is built based upon the vehicle scheduling method proposed by Wren and Holiday (1972). This approach operates by dividing the search area into sectors that are determined by two concentric circles and an arc of known length. Sectors are varied by changing the two radii of the circles and by rotating the arcs. A route is built when all nodes in a particular sector have been visited, or it is impossible to visit any other node of the same circle without violating the T_{\max} constraint.

Golden, Levy and Vohra (1987) propose an iterative heuristic for the OP which consists of three steps: route construction using a greedy method, route improvement using a 2-opt swap, and center-of-gravity which guides the next search step. Golden, Storchi and Levy (1986) present an approach that is divided in three steps: initial route selection that is performed by using the center of the gravity procedure, then the node insertion and node deletion procedure.

Golden, Wang and Liu (1988) combine Tsiligirides's S-algorithm concept, the center of gravity, and learning capabilities into another approach to solve the OP to provide probabilities for node selection. Keller (1989) uses his algorithm for the multi-objective vending problem to solve the OP. A path construction phase uses a measure identical to that of the S-algorithm. This is followed by a three steps improvement phase that uses node insertion and identification of node clusters. Wang, et al. (1995) propose an artificial neural network approach to solve the OP. A Hopfield-like neural network is formulated and a fourth order convex energy function is devised.

Ramesh and Brown (1991) propose a four-phase heuristic for the generalized orienteering problem, i.e., the cost function is not limited to a Euclidean function. The four phases consist of nodeinsertion; cost improvement, node deletion and maximal insertions. The route is improved by a 2-opt procedure followed by a 3-opt procedure in the second phase. In the third phase, one node is removed from the current route and one node is then inserted in an attempt to decrease the length of the route. Finally, as many unassigned nodes as possible are inserted onto the current route in order to increase the total score.

Chao, et al. (1996) introduce a two-step heuristic to solve the OP. In the first step, initialization, by using the starting and ending nodes as the two foci of an ellipse and the max T constraint as the length of the major axis, several routes are generated and the one with the highest score is the initial solution. The initial route is then improved by a 2-node exchange in the cheapest-cost way, and then improved by a 1-node improvement that tries to increase the total score.

Golden and Silberholz (2009), propose two parameters iterative algorithms approach to solve the

generalized orienteering problem. There are three phases: initialization in which the path is constructed and optimized by a 2-opt procedure, path tightening that is a local-search method that adds nodes to a solution when its length is less than the length limit, increasing that solution's score as much as possible and iterative modification in which the current solution is modified.

The Orienteering Problem (OP) is used to model single day itineraries. On the other hand, additional constraints for the itinerary, such as maximum budget to spend, or maximum number of POIs of certain type to be visited could be considered. Garcia et al. (2009) extend ILS (Iterated Local Search) to solve the Multi Constraint TOPTW, which allows to modeling the constraints related the maximum allowed budget to spent and maximum allowed POIs of certain type or category to visit. Souffriau et al. (2011) introduced a web-based tourist decision support system. The system uses the Arc Orienteering Problem (AOP), which is a single tour arc routing problem with profits. In AOP the arcs are associated with profits and travel costs and the goal is to find a route from a starting node to an end node with maximum profit and total travel cost not higher than a given value. Souffriau et al. use the AOP to solve the problem of planning cycle trips in the province of East Flanders. Their solution approach is based on a Greedy Randomized Adaptive Search Procedure (GRASP). This web-based system is extended with an SMS service that provides cyclists "in the field" with routes on demand.

The City Trip Planner introduced by Vansteerwegen et al. (2011) is a web-based tourist expert system that proposes custom-made city trips, tailored to the user's interests and context. The system is implemented as a web site. This system uses a meta-heuristic iterative Greedy Randomized Adaptive Search method. For each interaction, a list of possible visits is generated from an initial solution, which contains only the start and end of each tour. Those visits that have a heuristic value below a certain threshold are eliminated. A random visit from the remaining list is selected and applied to the current solution. The algorithm maximizes the tourist's trip satisfaction, by selecting the most interesting visits while respecting all constraints.

Černá et al. (2014) introduced a new problem in Combinatorial Optimization, namely the Most Attractive Cycle Tourist Path Problem (MACTPP), modeling the design of an origin-destination path with maximum attractiveness, subject to budget and duration constraints. The objective is to maximize the reward accumulated by traversing the arcs and the nodes of the path. These can be traversed several times, each time with a different satisfaction. Černá proposed an ILP model, which was solved by commercial software, with a dynamic generation of violated constraints.

Although the OP is studied from several researchers only during the recent years some authors proposed a genetic approach. With respect to heuristic or meta-heuristic approach, the Genetic Algorithm is generally able to explore a greater solutions' number. Good

results are given by this approach in several field, the major critic to this method is relative to computational time. Tasgetiren and Smith (2000) propose a Genetic Algorithm (GA) to solve the orienteering problem. A permutation representation is used and a penalty function is employed to help search the infeasible region. Four test sets are used. Tasgetiren's results are competitive, though the computational time is relatively high. Romero et al. (2012) propose a bi-level optimization model for the optimal location of bike-sharing stations, using a genetic algorithm, where the goal is to maximize the number of travelers that use the system. Its lower level is a modal split and assignment model, capable of reflection the interactions between car and bicycle mode. Time-reducing strategies applied to the genetic algorithm methods are valid, because the program returns similar solutions, with great computing time-savings. Finally De Falco et al. (2015) develop an evolutionary algorithm to generate personalized multiple-day itineraries with time windows using a linear chromosome encoding. Most recently other different metaheuristics are proposed to solve the OP as Ant Colony Optimization. Liang et al. (2006) develop and compare an ant colony optimization approach and a tabu search algorithm. Then Schilde et al. (2009) developed a multi-objective solution approach, which is a variant of the OP. Their approach outperforms the five-step heuristic of Chao et al. (1996), which deals with single objective OP instances. They developed a Pareto ant colony optimization algorithm and a multi-objective variable neighborhood search algorithm, both hybridized with path relinking. With respect to actual state of art in this paper we propose a Genetic Algorithm with a particular chromosome encoding that introduce the concept of "closed loop" for the chromosome's structure. This new configuration explores in more ample way the solution space obtaining good results in a short computational time.

3. PROBLEM DESCRIPTION

In this work, as mentioned in the introductory paragraph, the authors face the Tourist Trip Design Problem (TTDP) as an Orienteering Problem (OP). In the OP, a set of N location corresponding with nodes i is given, each with a score S_i . The starting point and the end point are fixed. The time t_{ij} needed to travel from vertex i to j is known for all vertices. The problem became more complex because not all the vertices can be visited since a given threshold T_{max} limits the available time. The goal of the OP is to determine a path, limited by T_{max} that visits some of the vertices, in order to maximise the total collected score. The scores are assumed to be entirely additive and each vertex can be visited at most once.

The orienteering problem can be formulated as follows: $S_i \geq 0$ is the score associated to node i , c_{ij} is the cost associated to path between node i and node j . Usually n nodes are considered in the Euclidean plane. Since the distance and travel time between nodes are determined by the geographical measure, distance is used as the

representative of path's cost. Generally, the mathematical model of the OP is formulated as follows:

$$Max \sum_{i=2}^{n-1} \sum_{j=2}^n S_i x_{ij} \quad (1)$$

Subject to

$$\sum_{j=2}^n x_{1j} = \sum_{i=1}^{n-1} x_{in} = 1 \quad (2)$$

$$\sum_{i=1}^{n-1} x_{ik} = \sum_{j=2}^n x_{kj} \leq 1 \quad \forall k=2, \dots, n-1 \quad (3)$$

$$\sum_{i=1}^{n-1} \sum_{j=2}^n c_{ij} x_{ij} \leq T_{max} \quad (4)$$

$$2 \leq u_i \leq n \quad \forall i=2, \dots, n \quad (5)$$

$$u_i - u_j + 1 \leq (n-1)(1 - x_{ij}) \quad \forall i, j=1, \dots, n \quad (6)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j=1, \dots, n \quad (7)$$

Where (1) represents the problem's Objective Function to maximize as total collected score S . Constraint (2) guarantee that the path starts in vertex 1 and ends in vertex N . Constraint (3) ensures the connectivity of the path and guarantee that every vertex is visited at most once. Constraint (4) ensures the limited time budget T . Constraints (5) and (6) are necessary to prevent sub-tours. Constraints (7) requires that the variables are binary. The decisional variable x_{ij} is equal to 1 if the node j to i are connected, 0 else. This formulation guarantee as result a tour able to:

- Visit as many PoIs as possible as sub-number of nodes;
- Visit PoIs at most once;
- Visit PoIs that maximize the Total Score (Objective Function)
- Visit PoIs connected among them
- Visit PoIs respecting the limitation time.

4. SOLUTION APPROACH

The concept of Genetic Algorithm (GA) was developed by Holland (1975) and then described by Goldberg (1989). GA are stochastic research techniques based on the mechanism of natural selection proposed by Darwin's theory: the strongest species has the greater opportunity to pass their genes to future generations via reproduction. Sometimes particular crossover procreates an original individual that start a new generation of individuals better than the old one. In the GA this randomness is opportunely reproduced to explore in a larger way the solution research area.

As reassumed in Askin et al. (2013), GA starts initializing a First Population composed by a pre-determinate number of individuals. Each individual is characterized by several elements such a chromosome and a fitness value. The individual's chromosome is composed by several gene that can assume binary or integer value, the combination of these genes usually

represents the problem's solution thanks to a correspondence between the gene's value and the problem's variables value. The individual's fitness value corresponds to solution goodness, and it's basic to determinate the individual's probability to survive at evolution process, so that bad individuals or bad problem's solutions are destined to not have a long life into population. The difference between the old and the new population in the evolutionary process is guaranteed by the presence of two important and always applied operations called Mutation and Crossover. After several generations, the best solution converges, and it hopefully represents the optimum or suboptimum solution to the problem at hand. GA are successfully applied to different contexts in order to solve a very large number of problems, for this purpose various and original genetic operators are developed, that revisit the original concept of mutation and crossover. Despite that, there are not yet many works in literature that have applied GA for solving Orienteering Problem especially in the last years. Tasgetiren and Smith (2000) is one of the lasts significant works in this field.

4.1. Encoding and Decoding

Encoding. In this work the authors propose the following encode of chromosome. Each chromosome is encoded as a vector of integer number representing the PoIs of our problem or the nodes, as we will call them from now on. In our case, the problem's solution is not represented by whole chromosome, but only of a part of it. The chromosome's part or as we call it, the tour, is composed by a sub-set of nodes that can be visited by tourist respecting the limitation time imposed. As showed in the Figure 1 we suppose that in each chromosome is contained one or more admissible tour for our Orienteering Problem.

In more details, our chromosome is composed of all nodes that the tourist wants to visit, without the start node and the end node. This choice depend on the following consideration:

- the start and end nodes are defined by the problem and they have to be inserted into each tour;
- the start and the end nodes don't have an associated score so that the goodness of solution is independent from them.
- The start and the end nodes are added to solution in a second step in order to calculate the effective length of tour in terms of time.

This coding of chromosome with respect to problem's solution, represented by a single tour, allows to have only admissible solutions, better or worst on the basis of Total score reached by visiting more nodes or better nodes.

Decoding

The original contribution of this paper consists into use a closed loop structure for the Chromosome. In this way it is possible to change the starting point of tour and to

explore a great number of solutions for each chromosome. In Figure 1 a usually linear structure of chromosome is given. In the Figure 2 the same chromosome is represented in a closed loop structure. The numbers inside of the chromosome represent the PoIs or nodes to visit on the basis or visiting order. For example in the Figure 1 the first tour is given by nodes "2", "1", "6", "5", "4". Wanting to find only 1 tour we identify the objective function value of chromosome as the objective function value of the best tour found inside the chromosome. Thanks to this assumption the research of tour inside the chromosome takes on a great importance. For this reason we introduce a closed structure.

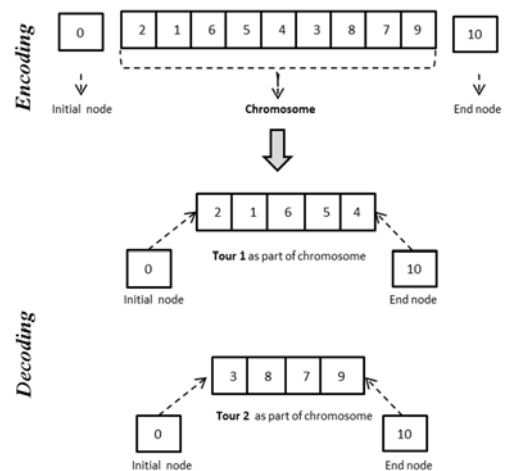


Figure 1: Linear Chromosome Structure code

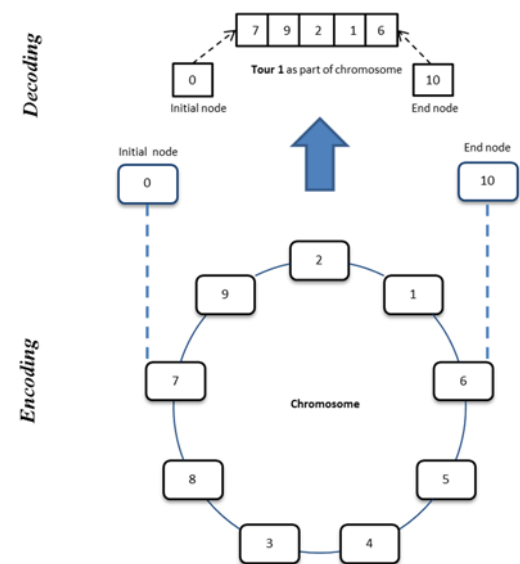


Figure 2: Closed loop Structure of Chromosome code

Thanks to this structure also the tour containing nodes at the initial and the end of chromosome can be considered in one same tour. In this way the solutions founded and evaluated for each chromosome are many and it is necessary a little number of interactions to found a good solution.

4.2. GA structure and Genetic operator

The general structure of the genetic algorithm, we have accomplished in this work is described below. In this application the individual is an object containing these information: chromosome, fitness function, objective function's value, the first and the last tour's node. Thanks to this structure, for each individual we can evaluate several tours and to obtain good problem's solution also with small population's sizes.

Genetic Algorithm

Parameters Setting

Population size PS
Number of nodes (Chromosome's length) n
Best Solutions Containers (BSC) size B
Frequency rate of genetic operators
Frequency rate of parents' selection from BSC
Number of interactions i (as terminal condition)
Maximum time T_{max} for visiting nodes
Set Pop (0) empty
Set BSC (0) empty

Initialization

- Generate a feasible solution randomly as individual
- Save it into the population Pop (i)
- Compare it with individual inside the BSC
 - If a worst solution is inside the BSC then
 - Update the BSC inserting the generated Individual and canceling the worst solution
- Loop until the population's size PS is reached
- Calculate Fitness Function for each individuals of Pop and BSC

While the number of iteration is not equal to predefined value of iterations do

Set iteration $i = 0$

Selection

Roulette function for selecting two individual as first and second parent sometimes from Pop and sometimes from BSC, on basis of Frequency rate of parents' selection from BSC

Crossover

- Apply PMX operator with frequency rate defined in parameters setting to obtain two new individuals as offspring
- Apply Single Crossover operator with frequency rate defined in parameters setting to obtain two new individuals as offspring
- Chromosome decode as n tours
- Evaluate Fitness Function
- Evaluate the BSC's updating

Mutation

- Apply Smart swap operator with frequency rate defined in parameters setting to obtain two offspring as mutation of individuals generated at Crossover level
- Chromosome decode as n tours
- Evaluate Fitness Function
- Evaluate the BSC's updating

Build

- Build new Population Pop (i+1) as collection of individuals generated with Crossover and Mutations Operation

If there are 3 iterations with the same best Individual, insert into Pop 20% of random Individual

Next i

End

Return the best solution

Return the best individual containing the best tour as solution of Orienteering Problem

The GA, typically uses two kind of genetic operators, Mutation and Crossover, in order to increase the probability to explore a greater area of solution with respect to others heuristic algorithms as Local search, Tabu search and Greedy. On the basis of this kind of problems and above all on the basis of its complexity, these operators can be used in an original way or in an original combination. In particular we have chosen to use 4 kind of operators:

- Single Crossover
- PMX Crossover
- Smart Swap
- 2-opt

Thanks to a probability definition, it is possible to test the effect of each single operator on chromosome. Generally the three operators are structured in way that a chromosome can be subject to all operators.

4.2.1. Single Crossover

Single Crossover operator proceeds in two steps, firstly two parents are selected randomly from population(i) – with I we represent the algorithm's iteration – secondly each pair of individual chosen is combined to obtain two offsprings. The ways to combine the genes of parents' chromosome are many and one can be preferable to another mainly on the basis of chromosome code.

In this work we have chosen two kinds of Crossover: Single and one known as Partially Matched Crossover PMX. In Medes (2013) is well explained as with the Single Crossover operation, an integer position x along the chromosome is randomly selected. The offspring is created swapping all the genes between $x+1$ and l as chromosome's length.

4.2.2. Partially-Matched Crossover (PMX Crossover)

PMX Crossover is the second one crossover operator used for our genetic algorithm. Basically, like for Single Crossover we select randomly two parents that will create two offsprings containing part of the parents' chromosome.

In this case, two integer positions x_1 and x_2 along the chromosome are randomly selected. The two integers are used as indexes to select the part of chromosome to

transfer from one parent (for example the mother) to child, the rest of child's chromosome will transfer to it from the other parent (for example the father). To avoid genes' redoubling a swapping technique is reproduced. The PMX – crossover operator is explained in the Appendix A Figure 5.

4.2.3. Smart Swap

Smart Swap is a variation of Random Swap Operation (RSO). RSO is a well-known operator of Mutation. The mutation operators have the task to preserve the diversification into solution's area search. In the Random Swap, two genes of father's chromosome are randomly selected and then exchanged to generate the offspring's chromosome. As a difference with respect to the crossover, applying this operator it isn't necessary a couple of individuals but just one.

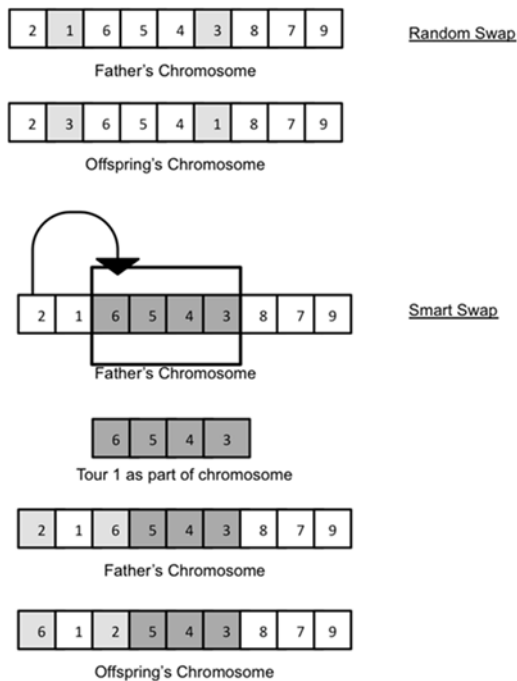


Figure 3: Smart Swap

For the structure of our chromosome code, in our work, it is necessary to apply a Smart Swap, if infact the two genes are selected into the best tour of chromosome, the total score does not change. This means that we don't generate enough variation to individuals' population. To overcome this limit we impose to select one of the two genes outside the best tour of chromosome.

Thanks to this order, we can evaluate more different solutions with different total score. Figure 3 represents the most important difference between the Random Swap and the designed Smart Swap.

4.2.4. 2-opt Mutation

2-opt is a simple local search algorithm first proposed by Croes in 1958 to solve the traveling

salesman problem. The main idea behind it, is to take a route that crosses over itself and reorder it so that this does not happen. It compares every possible valid combination of the swapping mechanism and gives good results in several routing problem as Vehicle Routing Problem (VRP) but also Capacited Vehicle Routing Problem (CVRP).

4.2.5. Elitism

A variant of the general process of constructing a new population is to generate a separate container called Best Solution Container (BSC) that contains the best solutions founded during the last iterations. When a new population is generated, the parents are selected through a probability, a part from old population and the other part from BSC. This strategy is known as *elitist selection* and guarantees that the solution quality obtained by the GA will not decrease from one generation to the next.

5. EXPERIMENTAL RESULTS

To evaluate the performance of the algorithm, we have proceeded firstly testing it with instances known in literature Chao (1996). Subsequently the algorithm has been implemented to a real case in the Naples town and the results are presented below.

5.1. Algorithm's validation

To validate the developed genetic algorithm and solve an orienteering problem for bike sharing tourist, we test it with the instances data Set 2 proposed by Tsiligrirides (1984).

The Appendix A shows our validation results and compares them with the results of others heuristic method used to solve an orienteering problem. Our approach is indexed with GA*, another GA is proposed in Tasgetiren (2000), NN (neural network) is proposed in Wang (1995), C (heuristic) is proposed in Chao (1996) e T (stochastic algorithm) is proposed in Tasgetiren (2000). We obtain for each instance the same value of objective function with respect to the others approaches; we don't obtain always the same sequence of visited PoIs but different sequence with the same value of objective function and in one case we have the same value of objective value but a better value (lower value) in terms of tour length (L_{path}). This result can be a starting point for a multi-objective optimization useful to take into consideration not only the Score of PoI but also, for example, the arrival time. However, our goal to validate the algorithm has been achieved thanks to this application.

The Table 1 and Figure 4 reassume the test realized with this data set known in literature as Set 2 of Tsiligrirides (1984). In the grey row we report the results obtained by Tasgetiren (2000) with a Genetic Algorithms, in the corresponding white row, we present our results for the same data. Thanks to closed structure of chromosome, the algorithm found good solution after a little number of iterations and with a computational time of the order of few seconds.

Table 1: Comparison of results on Tsiligirides's data set 2: objective function value.

Tmax	T	C	NN	GA	GA*	L_path
15	120	120	120	120		14,56
15					120	14,40
20	190	200	200	200		19,88
20					200	19,88
23	205	210	205	210		22,65
23					210	22,99
25	230	230	230	230		24,13
25					230	24,13
27	230	230	230	230		24,13
27					230	26,98
30	250	265	265	265		29,85
30					265	29,85
32	275	300	300	300		31,63
32					300	31,63
35	315	320	320	320		34,51
35					320	34,51
38	355	360	360	360		37,84
38					360	37,84
40	395	395	395	395		39,78
40					395	39,78
45	430	450	450	450		44,44
45					450	44,44

Figure 4: Comparison of results on Tsiligirides's data set 2: tours

Tmax	Tour
15	1 5 6 7 12 11 10 13 14 21
15	1 7 5 6 12 11 10 14 13 21
20	1 12 7 6 5 3 2 8 9 10 11 13 14 21
20	1 12 7 6 5 3 2 8 9 10 11 13 14 21
23	1 7 6 5 4 3 2 8 9 10 11 14 21
23	1 6 5 4 3 2 8 9 10 11 13 14 21
25	1 12 7 6 5 4 3 2 8 9 10 11 13 14 21
25	1 12 7 6 5 4 3 2 8 9 10 11 13 14 21
27	1 12 7 6 5 4 3 2 8 9 10 11 13 14 21
27	1 12 14 13 11 10 9 8 2 3 4 5 6 7 21
30	1 7 6 2 8 17 16 15 9 10 11 13 14 21
30	1 7 6 2 8 17 16 15 9 10 11 13 14 21
32	1 7 6 5 3 2 8 17 16 15 9 10 11 13 14 21
32	1 7 6 5 3 2 8 17 16 15 9 10 11 13 14 21
35	1 7 6 5 3 4 20 19 18 17 9 10 11 13 14 21
35	1 7 6 5 3 4 20 19 18 17 9 10 11 13 14 21
38	1 7 6 5 2 3 4 20 19 18 17 8 9 10 11 13 14 21
38	1 7 6 5 2 3 4 20 19 18 17 8 9 10 11 13 14 21
40	1 7 6 5 3 4 20 19 18 16 15 17 8 9 10 11 13 21
40	1 7 6 5 3 4 20 19 18 16 15 17 8 9 10 11 13 21
45	1 12 7 6 5 2 3 4 20 19 18 16 15 17 8 9 10 11 13 14 21
45	1 12 7 6 5 2 3 4 20 19 18 16 15 17 8 9 10 11 13 14 21

5.2. Real Application

In this paragraph we propose the results obtained for the bike sharing tourist problem in the city of Naples. We consider 30 PoIs, all included into urban areas of the city. We fix the PoIs of departures and of arrival and the maximum time that the tourist has to visit the city, variable from 20 minutes to 100 minutes.

We calculated the distance matrix and set the algorithm parameters (numbers of individuals, population and BSC's size, Frequency rate of genetic operators, frequency rate of genetic operators) on the basis of preliminary tests that have given the best configuration of algorithm.

After a great number of tests realized on instances known in literature and cited by Chao et al., we reached the configuration used for this real application. For example on Set 2 of Tasgetiren, tests reported in Table 1, the best average of objective function founded on 10 tests for the same instance. As Population's size we chose to use a number equal to $2n$ with n the number of nodes or PoIs. This value has been chosen because is a size that allows the presence of few couple of identical individuals. After several tests and observations, we reached this configuration:

- Population's size: $2n$
- Probability to Crossover: 0.8
- Probability to Single Crossover: 0.8
- Probability to PMX Crossover: 0.2
- Probability to Mutation: 0.4
- Probability to Smart Mutation: 0.6
- Probability to 2-opt: 0.4
- Probability to have a parent for elite: 0.1

Table 2: Real Application results

Tests	Iteration's Number	T _{max}	F.O.
1.a	100	20	30
1.b	200	20	30
1.c	500	20	30
1.d	1000	20	30
2.a	100	40	70
2.b	200	40	75
2.c	500	40	75
2.d	1000	40	75
3.a	100	60	95
3.b	200	60	100
3.c	500	60	105
3.d	1000	60	110
4.a	100	80	120
4.b	200	80	130
4.c	500	80	130
4.d	1000	80	130
5.a	100	100	145
5.b	200	100	150
5.c	500	100	160
5.d	1000	100	160

At this point we change for each instance of the problem only the number of iterations to find a better solution for the real application studied. The obtained results are described in Table 3. The first column represents the test's number, the second one the maximum time in which the tourist can visit the city (T_{max}), the third the objective function of founded solution. L_Path represents the time for the tour and the last column contains the iteration's number. In the Figure 5, the tour is represented as sequence of nodes (or PoIs), that the tourist will visit into the available time.

Table 3: Details of tours for real application's instances: objective function value.

Test	Tmax	Ga*	L_Path	Iteration's Number
1.a	20	30	19	100
2.d	40	75	40	200
3.g	60	110	60	1000
4.a	80	130	80	200
5.e	100	160	98	500

Figure 5: Details of tours for real application's instances: tours.

Tests	Tour																		
1	0	19	2	5	3	29													
2	0	10	26	20	16	28	3	5	2	29									
3	0	3	5	2	28	15	8	16	1	26	20	10	19	29					
4	0	19	2	5	28	15	26	20	16	10	1	12	9	3	13	29			
5	0	10	1	20	16	15	28	2	3	9	14	17	23	21	4	13	29		

The following figures show, as the goodness of solution is directly proportional to number of iterations but not always. In fact, for small instances with low T_{max} , a great number of interaction don't increase the solution founded with few iterations, for this reason a small calculation time (3-4 seconds) is enough to obtain the best solution of the problem. This is not true for greater instances, for example with $T_{max}=80$, there is an important difference between the solutions founded with 100 iterations respecting to solutions founded with 1000 iterations. The figures represent all results founded for every T_{max} and every number of iterations considered. For each test we run 10 times the algorithm with the aim to compare the best and the most frequent value of objective function founded. In both cases the curve of results is crescent with respect to the number of iterations.

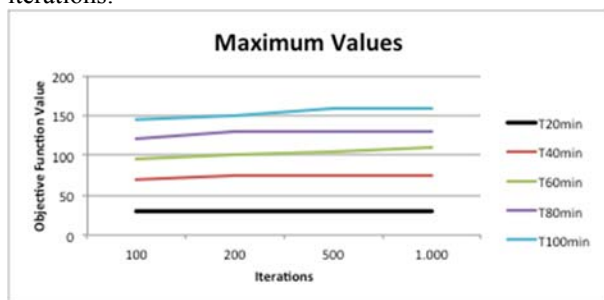


Figure 6: Maximum Value of Objective Function

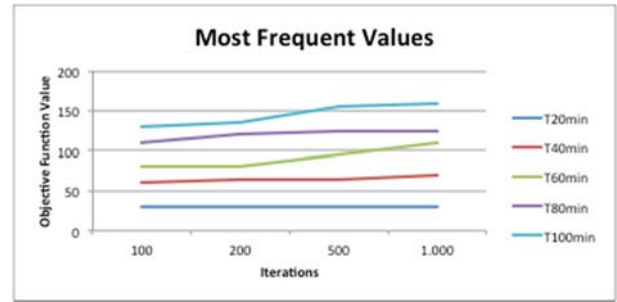


Figure 7: Most Frequent Value of Objective Function

The GA is coded in Visual Basic and implemented on Mac OS X Version 10.6.8. Processor 1.86 GHz Intel Core 2 Duo 2 GB 1067 MHz DDR3. We choose Visual Basic because it is easily translatable in others web oriented programming languages.

6. CONCLUSION

Actually the greater challenge in this research field is to propose systems able to: (i) give good solution in the shortest time possible (few seconds); (ii) react to dynamic reality of environment; (iii) interact with other mobility service in order to plan a greater trip in or out of the city. The systems developed have to be usable on smartphone, tablet, or pc and online or outline. The usefulness of this service system is high both for user as much as for the service manager. Thanks to these solutions it will be possible to know in advance the number of people that will visit a point of interest and then to organize the staff for the reception. On the other side it will be possible to make inaccessible a point of interest before planning the trip so that the tourist can maximize its satisfaction in the pre-determinate time for visiting. Future researches have to consider the possibility of tours' definition taking into account information about the amount of resources like battery or hydrogen of bikes. In this case the problem changes and the maximization of utility is subject to time respect and fuel availability. This kind of problem will be very useful to face another important question of bike sharing system's manager, the position of docking stations with respect to more popular tours.

REFERENCES

- Askin R.G., Baffo I., Xia M. 2013. Research Multi-commodity warehouse location and distribution planning with inventory consideration. International Journal of Production 06/2013, DOI: 10.1080/00207543.2013.787171
- Chao, I. M., Golden B. L., Wasil E.A.. 1996. A Fast and Effective Heuristic for the Orienteering. European Journal of Operational Research, vol. 88, pp. 475-489.
- Černá, A., Černý, J., Malucelli, F., Nonato, M., Polena, L., & Giovannini, A. 2014. Designing Optimal Routes for Cycle-tourists. Transportation Research Procedia, 3, 856-865.

- De Falco I., Scafuri U., Tarantino E., 2015. A Multiobjective Evolutionary Algorithm for Personalized Tours in Street Networks. *EvoApplications*. LNCS 9028, pp.115-127, 2015 DOI: 10.1007/978-3-319-16549-3_10
- Garcia, A., Vansteenwegen, P., Souffriau, W., Arbelaitz, O., & Linaza, M. 2009. Solving multi constrained team orienteering problems to generate tourist routes.
- Golden, B. L., Levy L., Vohra R., 1987. The Orienteering Problem. *Naval Research Logistics*, vol. 34, pp. 307-318.
- Golden B., Storchi G., Levy L., 1986. A Time-Relaxed Version of the Orienteering Problem- Proceedings of 1986 Southeast TIMS Conference (J.A. Pope and A. Ardalan, eds.), Myrtle Beach, pp. 35-37.
- Golden, B. L., Wang Q., Liu L., 1988 A Multifaceted Heuristic for the Orienteering Problem. *Naval Research Logistics*, vol. 35, pp. 359-366.
- Keller P. C., 1989. Algorithms to Solve the Orienteering Problem: A Comparison. *European Journal of Operational Research*, vol. 41, pp. 224-231.
- Liang Y. C., Smith A. E. 2006. An ant colony approach to the orienteering problem. *Journal of the Chinese Institute of Industrial Engineers*, 23(5), 403-414.
- Mendes J.M. (2013). A Comparative Study of Crossover Operators for Genetic Algorithms to Solve the Job Shop Scheduling Problem. *WSEAS TRANSACTIONS on COMPUTERS*, Issue 4, Volume 12, April 2013, 164-173
- Ramesh R., Brown K. M., 1991. An Efficient Four-Phase Heuristic for the Generalized Orienteering Problem. *Computers and Operations Research*, vol. 18, no. 2, pp. 151-165.
- Romero, J. P., Ibeas, A., Moura, J. L., Benavente, J., & Alonso, B. 2012. A simulation-optimization approach to design efficient systems of bike-sharing. *Procedia-Social and Behavioral Sciences*, 54, 646-655.
- Schilde, M., Doerner, K., Hartl, R., Kiechle, G., 2009. Metaheuristics for the bi-objective orienteering problem. *Swarm Intelligence* 3, pp. 179–201.
- Silberholz, J., & Golden, B. 2010. The effective application of a new approach to the generalized orienteering problem. *Journal of Heuristics*, 16(3), 393-415.
- Souffriau, W., Vansteenwegen, P., Berghe, G. V., & Van Oudheusden, D. 2011. The planning of cycle trips in the province of East Flanders. *Omega*, 39 (2), 209-213.
- Souffriau, W., Vansteenwegen, P., & Van Oudheusden, D.(2011) The orienteering problem: A survey. *European Journal of Operational Research* 209, 1-10.
- Tasgetiren, M. F., Smith A. E., 2000. A Genetic Algorithm for the Orienteering Problem. *Proceedings of the 2000 Congress on Evolutionary Computation*, San Diego, CA, pp. 1190-1195.
- Tsiligirides, T., 1984. Heuristic Methods Applied to Orienteering. *Journal of Operational Research Society*, vol. 35, no. 9, pp. 797-809.
- Vansteenwegen, P., Souffriau, W., Berghe, G. V., & Van Oudheusden, D. 2011. The city trip planner: An expert system for tourists. *Expert Systems with Applications*, 38(6), 6540-6546.
- Wang Q., Sun X., Golden B. L., Jia J., 1995. Using Artificial Neural Networks to Solve the Orienteering Problem. *Annals of Operations Research*, vol. 61, pp. 111-120.
- Wren A., Holiday A., 1972. Computer Scheduling of Vehicles from One or More Depots to a Number of Delivery Points. *Operations Research Quarterly*, vol. 23, pp. 333-344.

AUTHORS BIOGRAPHY

Ilaria Baffo received her PhD in management engineering from Tor Vergata University in 2010. Her research interests include the area of Supply Chain Management, Decision Support System, Distributed Network Design, Operation Research Application, Optimization's Tools. She worked for 6 years with a public institute of research in Italy and now she applied her knowledge at Institute for Applied Mathematics (National Research Council). She has published two paper in an international journal and a variety in international conference proceedings. She is also involved in technology transfer from research field to industrial enterprises.

Pasquale Carotenuto, Technologist in the National Research Council (CNR) at the Institute for Applied Mathematics "M. Picone" since 2006, and at the CNR - Institute of Industrial Technologies and Automation from 2001 to 2006, and in the direction staff of the CNR - Transportation Project 2" from 1995 to 1999. Graduated in Civil Engineering on Transportation at University of Naples Federico II, he is co-teachers in a course on Design and Simulation of Production Systems and Services at the University of Rome "Tor Vergata".

Giovanni Storchi, at the 'University of Rome La Sapienza has taught courses in "Graph", "Networks Flow", "Data Structures and Complexity", he was a visiting scientist in Mexico, Canada, South Africa and USA, he is now Professor of "Routing", at the Master "Data Intelligence and Decision Making Strategies" at the Department of Statistics, University of Rome la Sapienza.

APPENDIX A

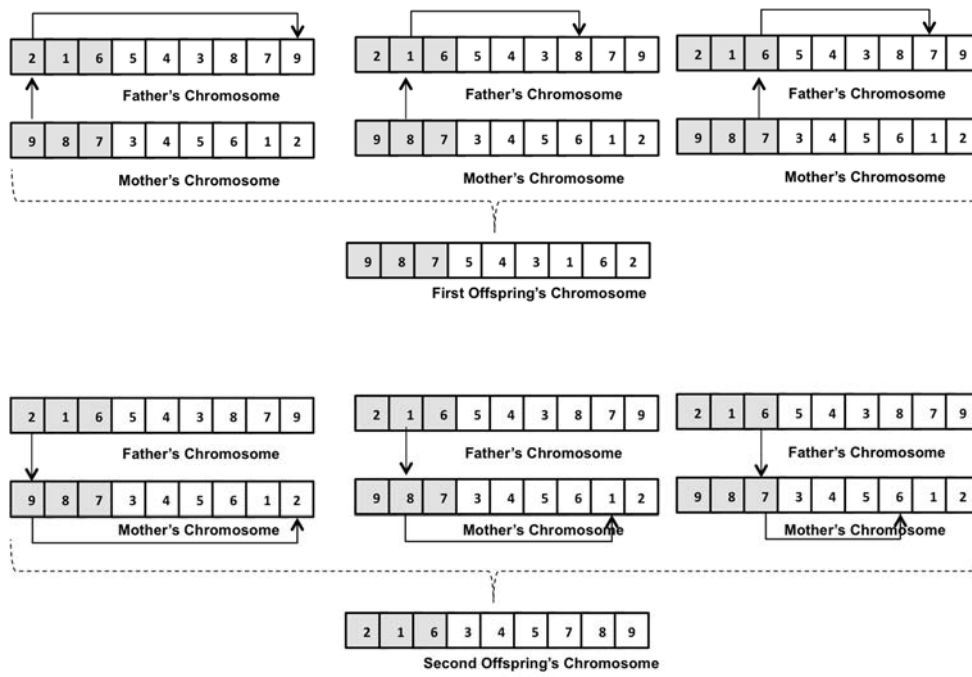


Figure 8: PMX-Crossover Operator