

Article



# Time-Dependent Theme Park Routing Problem by Partheno-Genetic Algorithm

Zhang Yang <sup>1,\*</sup>, Jiacheng Li<sup>2</sup> and Lei Li<sup>1</sup>

- <sup>1</sup> Department of Information Engineering, Faculty of Science and Engineering, Hosei University, Tokyo 184-8584, Japan; lilei@hosei.ac.jp
- <sup>2</sup> Department of Electrical, Electronics and Information Engineering, Faculty of Engineering, Kanagawa University, Yokohama 221-8686, Japan; lijiacheng@kanagawa-u.ac.jp
- \* Correspondence: yang.zhang.6j@stu.hosei.ac.jp; Tel.: +81-70-4134-6022

Received: 24 September 2020; Accepted: 2 December 2020; Published: 9 December 2020



**Abstract:** With the improvement of people's living standards and entertainment interests, theme parks have become one of the most popular holiday places. Many theme park websites provide a variety of information, according to which tourists can arrange their own schedules. However, most theme park websites usually have too much information, which makes it difficult for tourists to develop a tourism planning. Therefore, the theme park routing problem has attracted the attention of scholars. Based on the Traveling Salesman Problem (TSP) network, we propose a Time-Dependent Theme Park Routing Problem (TDTPRP), in which walking time is time-dependent, considering the degree of congestion and fatigue. The main goal is to maximize the number of attractions visited and satisfaction and to reduce queues and walking time. To verify the feasibility and the effectiveness of the model, we use the Partheno-Genetic Algorithm (PGA) and an improved Annealing Partheno-Genetic Algorithm (APGA) to solve the model in this paper. Then, in the experimental stage, we conducted two experiments, and the experimental data were divided into real-world problem instances and randomly generated problem instances. The results demonstrate that the parthenogenetic simulated annealing algorithm has better optimization ability than the general parthenogenetic algorithm when the data scale is expanded.

**Keywords:** tourism planning; theme park problem; routing problem; time-dependent; partheno-genetic algorithm

# 1. Introduction

With the improvement of people's living standards and material level, people's daily lives are no longer solely focused on the pursuit of food and clothing but include more spiritual pursuits. Holiday outings have become a common form of entertainment for modern people. Among them, theme parks have become the best place for short-term travel [1]. Some famous theme parks such as Tokyo Disneyland and Universal Studios Osaka attract millions of visitors every year, which shows that, as people's living standards improve, people's demand for the entertainment industry is also increasing. However, several issues in theme parks need to be resolved. For example, for popular attractions, tourists need to endure long queues, which may take two or three hours. In addition, since most theme parks are large in size and populated, tourists spend a lot of time walking and queuing. Therefore, quickly planning the best way to enjoy the theme park with limited time and energy, avoiding congestion, and enjoying as many attractions as possible has become the most important concerns for tourists. Moreover, for the managers of theme parks, the degree of satisfaction of tourists is fundamental to the sustainable development of the theme park. The higher the evaluation of tourists is, the more other tourists will be attracted to the theme park. In contrast, if the degree of satisfaction of tourists is very low, the development of the theme park will be restricted.

In 2003, the theme park problem was defined by Kawamura et al. [2], that is, in a theme park with multiple attractions, tourists visit the attractions as individuals or groups to minimize congestion and satisfaction. In addition, they developed a coordinated scheduling algorithm based on large-scale user support to solve this problem, proving that the average waiting time and congestion in theme parks can be reduced by guiding tourist groups, and the average satisfaction of group visitors can be improved. In short, the problem of theme parks is to maximize personal or crowd satisfaction using management rules and planned timetables. Considering the group users, Yasushi et al. [3] developed networks for theme park issues, such as small world networks and scale-free networks. Simulation experiment results show that congestion can be greatly relieved, and satisfaction can be further improved.

Most previous studies involving the theme park problem focus on large-scale travel scheduling for group users. However, differently, in this paper, we aim to help individual users avoid congestion and improve the degree of satisfaction of the visitors in the theme park. We can call this kind of problem theme park routing problem (TPRP). Thus far, route planning for individual users is usually divided into two categories:

- 1. One category aims to find an efficient route among the chosen attractions [4].
- 2. The other is to select some attractions to visit and maximize the obtained total score for visited attractions during a limited time, where origin and destination are appointed in advance [5].

The research in this paper belongs to the second category, which does not need to specify the attractions in advance. Tsai et al. [6] developed a route recommendation system where the recommended route satisfies visitor requirements using previous tourists' favorite experiences. Lee et al. [7] presented an ontological recommendation for a multi-agent for Tainan City travel, including a context decision agent and a travel route recommendation agent. Lim, K. et al. [8] proposed an algorithm called PersTour for recommending personalized tours using Points of Interest (POI) popularity and user interest preferences, which are automatically derived from real-life travel sequences based on geo-tagged photos. Mor, M. et al. [9] developed a bi-directional constrained pathfinder nearest neighbor route calculation algorithm to compute routes that visit the most popular touristic locations among photographers. Gionis, A. et al. [10] presented two alternative instantiations of a framework for generating customized tour recommendations as a paradigm of an intelligent urban navigation service. Sengupta, L. et al. [11] considered three different strategies for selecting the starting location and compared their effectiveness regarding optimizing tour length. Hirotaka et al. [12] proved the tour recommendation problem can be solved as the integer programming problem using a similar formulation as used in the traveling salesman problem (TSP). Matsuda et al. [13] established a simple model of the optimal sightseeing routing problem and solved the model with the exact algorithm and the heuristic algorithm, respectively.

From the perspective of mathematical modeling, our problem can be described as follows. A set of points is given, along with associated scores and a connecting network. Under this assumption, a path needs to be found between the specified starting point and end point to maximize the total score at a given time. Through the above description, we find that this kind of problem can be attributed to the TSP [14]. It should be noted that, due to the limited time, it is impossible to select all points, and some points should be discarded. Subsequently, this problem is also called the selective traveling salesperson problem (STSP) [15] or traveling salesman problem with profits [16], which is a generalized traveling salesman problem in which profit is associated with each vertex and only some vertices can be visited due to time constraints [17]. The STSP is also known as the orienteering problem [18] and the maximum collection problem [19]. As the STSP is an Non-Deterministic Polynomial (NP) hard problem, the exact algorithms are very time-consuming, thus most researchers focus on heuristic algorithms, such as the Tabu search (TS) heuristic algorithm [20,21] or the ant colony optimization (ACO) approach [22].

The major weakness of the previous research is that only static route networks are constructed, but the change of time that leads to a change in the next journey is not considered. Considering

this problem, Bouzarth et al. [23] set the service time as time-dependence but did not consider that the travel times between two vertices are stochastic functions that depend on the department time from the first vertex. Moreover, the previous studies are all single objective functions, and there are few cases of solving multi-objective problems at the same time. The algorithm research of routing optimization is mainly carried out under the condition of static networks, and there is less research using dynamic networks.

Unlike previous research, we first propose to minimize total walking and queuing time and maximize the number of attractions and the degree of satisfaction, which has a multi-objective function. Then, we consider the walking time of time dependence [24,25]. Second, we introduce two algorithms in detail to solve this problem. Finally, in the experimental stage, we use real-world problem instances and randomly generated problem instances and analyze the experimental results to prove the correctness of the model and the effectiveness of the algorithm.

#### 2. Time-Dependent Theme Park Routing Problem Based on Multi-Objectives

#### 2.1. Problem Description

Let G = (N, A) be a connected digraph with node set  $N = \{1, 2, ..., n\}$  and arc set  $A = \{(m, n) | m, n \in N, m \neq n\}$ , where node 1 is the starting point, and n is the end point. In the time-dependent theme park routing problem, each node represents an attraction in the theme park. What is associated with each node is a utility score and a function of dwelling time, which is related to arrival time, queuing time, and time at the attraction. When visitors enter from the designated entrance, they should find the most satisfactory route to the attraction and cannot leave the exit after the designated time *TD*. In this process, the objective of visitors is to find a route that starts from node 1 and ends at node n before *TD*, such that the total utility collected by all visited nodes in the route is maximized and the number of nodes experienced is maximized but the dwelling time is minimized.

Assume that the distance between the two attractions (m, n) is d, and the walking times that the visitors need to cover this section (m, n) in two connected time periods are  $T_1$  and  $T_2$ , respectively, and  $t_B$  is the boundary time of the two time periods. It should be noted that  $T_x$  stands for time period and  $t_x$  stands for time point. The walking time of the visitor in the section (m, n) is different for different time periods. The arrival time at node m is  $t_m$ , the queuing time at node m is  $\alpha_m$ , the time spent at the attraction at node m is  $\beta_m$ , and the departure time for node m is  $r_m$  ( $r_m = t_m + \alpha_m + \beta_m$ ). If  $r_m \le t_B - T_1$ , the walking time to cover this section in the preceding period is  $T_1$ ; if  $r_m \ge t_B$ , the whole walking process is completed in the following period, and the corresponding walking time is  $T_2$ . Only when  $t_B - T_1 < r_m < t_B$  is the walking time between  $T_1$  and  $T_2$ . Assume that the walking distances of the visitor in the preceding and the following time periods are  $d_1$  and  $d_2$ ,  $d_1$  corresponds to the travel distance of tourists in  $T_1$  period, and  $d_2$  corresponds to the travel distance of tourists in  $T_2$  period. The total walking time is  $d_1(\frac{T_1}{d}) + d_2(\frac{T_2}{d})$ . Subsequently, the time is  $t_B - r_m$  for visitors to cover  $d_1$ , and the time for visitors to cover  $d_2$  is:

$$\frac{d_2}{d}T_2 = \frac{d-d_1}{d}T_2 = \frac{d-\frac{d}{T_1}(t_B - r_m)}{d}T_2 = \left(1 - \frac{t_B - r_m}{T_1}\right)T_2$$

Therefore, the total walking time from the node *m* to the node *n* is  $t_B - r_m + (1 - \frac{t_B - r_m}{T_1})T_2$ , and the arrival time for node *n* is:

$$t_n = r_m + t_B - r_m + \left(1 - \frac{t_B - r_m}{T_1}\right)T_2 = t_B + \left(1 - \frac{t_B - r_m}{T_1}\right)T_2 = \frac{T_2}{T_1}r_m + \left(1 - \frac{T_2}{T_1}\right)t_B + T_2$$
(1)

As can be seen from the Equation (1), the time  $t_n$  for arriving at the node n is an increasing function of the departure time  $r_m$ .

Assume that the playing time in one day is divided into *P* time periods, then  $[t_{B,p}, t_{B,p+1}]$  represents the time period  $p(p \in [1, P])$ . After the visitor arrives at attraction *m*, the queuing time and the time spent at the attraction are allowed before the visitor departs from attraction *m*. Let  $T_{mn, p}$  be the walking time from node *m* to node *n* during period *p* (regardless of the period crossing), and let  $u_{mn, p}$  be the time that the visitor walks from node *m* to node *n*. For any section (*m*, *n*), there are two possibilities:

- 1. The visitor walks from node *m* to node *n* without crossing the time period p(p = 1, 2, ..., P);
- 2. The visitor walks from node *m* to node *n*, crossing from period *p* to period p + 1(p = 1, 2, ..., P 1).

The corresponding mathematical formulations are as follows:

$$u_{mn,p} = \begin{cases} T_{mn,p}, & r_m < t_{B, p+1} - T_{mn,p}, p = 1, 2, \dots, P\\ \left(\frac{T_{mn,p+1}}{T_{mn,p}} - 1\right)r_m + \left(1 - \frac{T_{mn,p+1}}{T_{mn,p}}\right)t_{B,p+1} + T_{mn,p+1}, t_{B,p+1} - T_{mn,p} \le r_m < t_{B,p+1}, p = 1, 2, \dots, P - 1\end{cases}$$

## 2.2. Model

Based on the questions raised, we consider the following assumptions:

- The queuing time of attractions is acquired according to the real queuing time of the attraction (e.g., Disney Resort and Universal Studios).
- The time spent at each attraction is given in advance.
- Routes exist between any two attractions.
- The preference of visitors for each attraction is pregiven.

The time-dependent theme park routing problem model contains the following parameters:

*N*: Set of attractions

*m*: Index of attraction *m*,  $m \in N$ 

TE: Entrance time

TD: Departure time

*P*: The number of time periods in one day

*p*: Index of the time period,  $p \in [1, P]$ 

 $c_1, c_2, c_3$ : Weights of the functions  $Z_1, Z_2, Z_3$ , in which the sum of three weights is equal to 1  $X_m$ : If attraction *m* is selected,  $X_m = 1$ ; otherwise,  $X_m = 0$ , which is a decision variable

 $Y_{mn,p}$ : The route from node *m* to node *n* with the visitor departs from node *m* in period *p* is selected, it takes 1; 0 otherwise

 $\alpha_m$ : Queuing time for attraction *m* 

 $\beta_m$ : Playing time spent at attraction *m* 

 $u_{mn,p}$ : Walking time of visitor departing node *m* in time period *p* to node *n* 

 $T_{mn,p}$ : Time for visitor walking from node *m* to node *n* in period *p* (regardless of the time period crossing)

 $r_m$ : Departure time for attraction m

 $t_m$ : Arrival time for attraction m

 $e_m$ : Utility that the customer obtained from node m

 $\lambda_1, \lambda_2, \lambda_3$ : The conversion factors of utility in term of cost

There are three objectives involved: maximize the number of attractions, maximize the satisfaction of the visitors, and minimize walking time and queuing time.

1. Most tourists want to enjoy as many attractions as possible in the theme park within a limited time. Therefore, we consider the maximum number of attractions visited in a day, and the calculation formula is as follows:

$$Z_1 = c_1 \sum_{m \in N} X_m$$

Mathematics 2020, 8, 2193

2. Most tourists hope to get as much satisfaction as possible in the theme park within a limited time. Therefore, we consider the maximum satisfaction of tourists in a day, the formula is as follows:

$$Z_2 = c_2 \sum_{m \in N} e_m X_m$$

3. For most tourists, they want to spend as much time as possible on each attraction, instead of moving on and queuing to find other attractions. Therefore, we considered the minimum walking time and the queuing time between attractions in a day; the calculation formula is as follows:

$$Z_3 = -c_3 \left( \sum_{m \in \mathbb{N}} \alpha_m X_m + \sum_{m,n \in \mathbb{N}} \sum_{p=1}^P u_{mn,p} Y_{mn,p} \right)$$

Finally, the overall mathematical model is established as follows:

$$\operatorname{Max} \lambda_1 Z_1 + \lambda_2 Z_2 + \lambda_3 Z_3 \tag{2}$$

$$TE + \sum_{m \in N} \alpha_m X_m + \sum_{m,n \in N} \sum_{p=1}^P u_{mn,p} Y_{mn,p} + \sum_{m \in N} \beta_m X_m \le TD$$
(3)

$$\sum_{\substack{m \in N \\ m \neq n}} \sum_{p=1}^{P} Y_{mn,p} = 1, n \in N$$
(4)

$$\sum_{\substack{n \in N \\ n \neq m}} \sum_{p=1}^{p} Y_{mn,p} = 1, m \in N$$
(5)

$$\sum_{\substack{m,n \in N \\ m \neq n}} \sum_{p=1}^{P} Y_{mn,p} \le |S| - 1, \ \forall S \in N, \ 2 \le |S| \le |N| - 1$$
(6)

$$X_m \in \{0, 1\}, \ m \in N,$$
 (7)

$$Y_{mn,p} \in \{0,1\}, m, n \in N, p \in [1,P]$$
(8)

The objective Equation (2) can maximize the attractiveness and the practical value (satisfaction) of the project and minimize queue time and walking time. Equation (3) ensures that the departure time from the theme park shall not exceed the preset departure time. Equations (4)–(6) are general constraints of TSP. The first two constraints restrict each scenic spot to be visited exactly once, and the last constraint is the sub-trip elimination constraint. Equations (7) and (8) are the domain of variables.

#### 3. Algorithm

In order to solve the model better, we use two different methods to compare.

#### 3.1. Parthenogenetic Algorithm (PGA)

The parthenogenetic algorithm (PGA) is an improved GA which was put forward by Li and Tong in 1999 [26]. Unlike classic GA, PGA performs its operations based on another chromosome other than two chromosomes and does not include a crossover operator. Due to its professionalism, PGA can successfully solve specific problems. Although PGA can overcome the "premature" problem of conventional GA, it is more suitable for handling chromosomes of different lengths.

The algorithm uses new genetic operators, namely permutation operators, shift operators, and inversion operators. They are characterized by simple genetic operations, which reduces the requirement for initial population diversity and avoids "premature convergence". Therefore, this algorithm is very suitable for solving optimization problems.

#### 3.1.1. Solution Representation

In PGA, each solution is represented by a chromosome, and each chromosome contains multiple variants, namely genes. Here, each gene represents the attraction of a theme park. Different arrangements of attractions require different solutions. In this study, chromosomes can be indicated as follows:

$$A = (c_1, c_2, c_3, \dots, c_{i-1}, c_i, c_{i+1}, \dots, c_{j-1}, c_j, c_{j+1}, \dots, c_n)$$

where A is a chromosome, and  $c_i (i \in \{1, ..., n\})$  expresses the gene.

#### 3.1.2. Evolution Operators

(1) Permutation operator. In this study, the permutation operator includes two modes: the first is a single-point translocation, where the positions of two genes on the same chromosome can be interchanged; the second is a multi-point translocation, where the positions of two genes on the same chromosome are exchanged. The replacement operators of the above two modes can generate new chromosomes according to the replacement probability. Examples of single-point transposition and multi-point transposition are as follows. Note that B and B' are produced by single-point and multiple-point translocation of A chromosome, respectively.

$$A = (c_1, c_2, c_3, \dots, c_{i-1}, c_i, c_{i+1}, \dots, c_{j-1}, c_j, c_{j+1}, \dots, c_n)$$

$$\Downarrow$$

$$B = (c_1, c_2, c_3, \dots, c_{i-1}, c_j, c_{i+1}, \dots, c_{j-1}, c_i, c_{j+1}, \dots, c_n)$$

$$B' = (c_1, c_2, c_3, \dots, c_{i-1}, c_j, c_{i+1}, \dots, c_n, c_i, c_{j+1}, \dots, c_{j-1})$$

(2) Shift operator. The shift operator refers to randomly selecting a substring in the chromosome according to the shift probability, the gene of the substring is shifted one bit backward, and the last gene is located at the first position of the substring. Similarly, the shift operator includes two shift modes: single-point shift and multi-point shift operation. In the example below, H is a chromosome including multiple genes  $k_i$  ( $i \in \{1, 2, ..., n\}$ ). I and I' are chromosomes obtained by performing single-point shift and multi-point shift operations on H, respectively.

(3) Inversion operator. The inversion operator refers to the process of continuously inverting the head and the end genes in the substring of the chromosome according to the inversion probability, wherein the selected substring and its length are randomly selected. Inversion operators can also be divided into single-point inversion and multi-point inversion. For example, assume M is a chromosome including multiple genes  $l_i$  ( $i \in \{1, 2, ..., n\}$ ). Based on the operations of single-point and multi-point inversions for chromosome M, we can obtain the chromosomes N and N'. The above procedures can be seen as follows.

$$M = (l_1, l_2, l_3, l_4, l_5, \dots, l_{i-1}, l_i, l_{i+1}, \dots, l_{j-1}, l_j, l_{j+1}, \dots, l_n) 
\downarrow 
N = (l_4, l_3, l_2, l_1, l_5, \dots, l_{i-1}, l_i, l_{i+1}, \dots, l_{j-1}, l_j, l_{j+1}, \dots, l_n) 
N' = (l_4, l_3, l_2, l_1, l_5, \dots, l_{i-1}, l_j, l_{j-1}, \dots, l_{i+1}, l_i, l_{j+1}, \dots, c_n)$$

#### 3.1.3. Algorithm Flow

The procedures of the parthenogenetic algorithm are summarized as follows:

Step 1: Coding. Serial number coding is used in parthenogenetic algorithm.

Step 2: Initialization. A feasible solution is randomly generated as the initial population.

Step 3: Fitness function. The fitness function is the evaluation standard of the path plan, which represents the survival ability of the genetic individual. Here, we set the objective function as the fitness function.

Step 4: Select. The resulting population is divided into several groups evenly. In this study, every four individuals formed a group. The best individuals in each group will be directly retained as the next generation population.

Step 5: Parthenon-genetic. Realize replacement, shift, and inversion operations. Among the remaining individuals, random methods are used to select genes and gene strings. After that, the three newly generated individuals in each group will be inherited to the next generation.

Step 6: Calculate the fitness of the newly generated population.

Step 7: Determine whether the termination conditions are met. When the maximum number of iterations is reached, go to step 8. Otherwise, go to step 4.

Step 8: Return the optimal solutions and stop the algorithm.

The flow chart is shown in Figure 1:



Figure 1. The flow chart of the parthenogenetic algorithm (PGA).

# 3.2. Annealing Parthenogenetic Algorithm (APGA)

The global optimization ability of genetic algorithm is strong, but the local optimization ability is insufficient. The simulated annealing algorithm is strong in local optimization but weak in global optimization ability. Therefore, the evolution mechanism of simulated annealing algorithm can be integrated into genetic algorithm to enhance its local optimization ability [27].

The basic idea of simulated annealing algorithm was originated from the physical annealing process in real life. The optimal solution is acquired by abstracting the process of cooling and heating isothermals from the real physical annealing process. The local optimization ability of the algorithm is ensured using a greedy strategy and its special Metropolis criterion [28].

Li Liu et al. [29] proposed a new method combining simulated annealing (SA) and genetic algorithm (GA) to solve the problem of bus route design and frequency setting for a given road network with fixed bus stops and driving demands.

The simulated annealing method has a fault-tolerant ability and can accept inferior solutions with a certain probability. The probability is called the probability of the acceptance of the new solution, and its degree is influenced by the current temperature and the fitness difference of new and old solutions [30]. The general trend is that the lower the temperature, the lower the probability of acceptance, the larger the difference, the lower the probability of acceptance. The above can be represented by a mathematical formula, as shown in Equation (9):

$$P = \begin{cases} 1 & E(x_{new}) < E(x_{old}) \\ \exp\left(-\frac{E(x_{new}) - E(x_{old})}{T}\right) & E(x_{new}) \ge E(x_{old}) \end{cases}$$
(9)

 $E(x_{new})$  refers to the fitness of the new chromosome, and  $E(x_{old})$  refers to the fitness of the individual parent chromosome. T is the current temperature, and an initial temperature and cooling coefficient are set at the beginning of the genetic operation. As the iteration proceeds, the initial temperature decreases continuously. At the end of the iteration, because the temperature is already very low, the probability of accepting the new solution is almost zero. A random number between 0 and 1 is generated after acquiring the probability of acceptance. A new solution is not accepted if the number is greater than the probability of acceptance *P* but accepted if it is smaller than *P*. The selection method of the probability of acceptance is also called the Metropolis criterion.

Metropolis criterion based on a simulated annealing algorithm certainly accepts inferior solutions while accepting elegant solutions so as to ensure population diversity and further avoid the possibility that the algorithm is stuck in the optimal local solution.

Therefore, after the new solution is generated by the evolution operation of genetic algorithm, the Metropolis criterion can be used to determine if the new solution is reserved; the optimization steps are as follows:

Step 1: Encoding. A serial number encoding is adopted in the parthenogenetic algorithm.

Step 2: Initialization. Generate feasible solutions as the initial population randomly.

Step 3: Fitness function. Fitness function is the evaluation criterion of the path scheme, and it represents the viability of genetic individuals. Here, we set the objective function as the fitness function.

Step 4: Selection. The resulting population is divided into several groups evenly. In this study, every four individuals formed a group. The best individuals in each group were directly retained as the next generation population.

Step 5: Parthenogenetic. Implement permutation, shift, and inversion operations. Among the remaining individuals, random methods are used to select genes and gene strings. Then, inherit the three newly generated individuals from each group to the next generation.

Step 6: Calculate the fitness of the newly generated population.

Step 7: Mix new and original populations. First, reserve the best individuals in the two populations and calculate the average fitness. Then, select one among the mixed population (except for the optimal individual) at random. If the individual is better than the average value, reserve it; otherwise, check whether it meets the M criterion. If satisfied, reserve it. If not, discard it. Operate in this way until the population of the quantity, which is the same as the original individuals, and then stop.

Step 8: Determine whether termination conditions are met. When the maximum iteration is reached, go to Step 8. Otherwise, go to Step 4.

Step 9: Return the optimal solutions and stop the algorithm.

The algorithm flow chart is shown in Figure 2:



Figure 2. The flow chart of annealing parthenogenetic algorithm (APGA).

#### 4. Discussion Computation Results

In the experiment, we use two kinds of theme parks with different scales as examples. First, we use the Tokyo Disney Sea with 28 attractions as a real-world problem to prove the effectiveness of the model and algorithm. Second, to prove the stability of the model and algorithm, we expand the scale of the experiment and randomly generate an example of a theme park with 60 attractions.

#### 4.1. Real World Problem Instances

First, the Tokyo Disney Sea with 28 attractions is selected as the test object to verify the correctness and the effectiveness of the proposed model and algorithm. In the network of attractions, 0 represents the entrance (exit), and the remaining are attractions. For this problem, we focus on a visitor leaving the entrance of the theme park, visiting the attractions, and then leaving the theme park. The purpose is to increase the satisfaction of visitors and increase the utilization of park resources. Table 1 illustrates the shortest walking distance between any two attractions. The time spent at the attraction, the queuing time, and the degree of satisfaction for each attraction are provided in Table 2, and the satisfaction for the attractions are generated from this interval [0,1]. In addition, the following assumptions are established:

- (i.) The entrance and the exit of the theme park are at the same location.
- (ii.) The walking speed of visitors can be obtained by a function  $y = 105.49 \times 10^{-0.002x}$  ( $x \ge 0$ ), where x represents time (in this experiment, x = 0 is 9 a.m., x = 30 is 9.30 a.m., and x = 180 is 12 p.m.), and y represents the corresponding speed.
- (iii.) The time visitors spent at each attraction is composed of three parts: walking time, playing time spent at the attractions, and queuing time.

\_

\_

28

1170

600

590

Attraction														A	ttractio	n													
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
0	0	570	1180	390	1140	1130	760	1450	1140	1330	2400	2410	2530	2820	2640	3210	2680	3430	3330	2780	2840	2980	2120	2430	2920	3000	2760	1170	1170
1	570	0	610	720	1470	1460	1090	1780	1470	1660	2730	2740	2860	2250	2070	2640	2110	2760	2660	2210	2270	2410	1550	1860	2350	2430	2190	600	600
2	1180	610	0	1330	2080	2070	1700	2390	2080	2270	3030	2840	2960	2240	2060	2630	2100	2750	2650	2200	2260	2400	1540	1850	2340	2420	2180	590	590
3	390	720	1330	0	990	980	610	1300	990	1180	2250	2260	2380	2970	2790	3360	2830	3540	3480	2930	2990	3130	2270	2580	3070	3150	2910	1320	1320
4	1140	1470	2080	990	0	190	380	310	200	190	1260	1270	1390	1870	2050	2260	2650	2440	2540	2750	2810	2950	2570	2260	2890	2970	2730	2070	2070
5	1130	1460	2070	980	190	0	370	500	190	380	1450	1460	1580	2060	2240	2450	2840	2630	2730	2940	3000	3140	2760	2450	3080	3160	2920	2060	2060
6	760	1090	1700	610	380	370	0	690	380	570	1640	1650	1770	2250	2430	2640	3030	2820	2920	3130	3190	3330	2640	2640	3270	3350	3110	1690	1690
7	1450	1780	2390	1300	310	500	690	0	510	300	950	960	1080	1560	1740	1950	2340	2130	2230	2440	2500	2640	2260	1950	2580	2660	2420	2380	2380
8	1140	1470	2080	990	200	190	380	510	0	390	1460	1470	1590	2070	2250	2460	2850	2640	2740	2950	3010	3150	2770	2460	3090	3170	2930	2070	2070
9	1330	1660	2270	1180	190	380	570	300	390	0	1250	1260	1380	1860	2040	2250	2640	2430	2530	2740	2800	2940	2560	2250	2880	2960	2720	2260	2260
10	2400	2730	3030	2250	1260	1450	1640	950	1460	1250	0	190	310	790	970	1180	1570	1360	1460	1670	1730	1870	1490	1180	1810	1890	1650	2440	2535
11	2410	2740	2840	2260	1270	1460	1650	960	1470	1260	190	0	120	600	780	990	1380	1170	1270	1480	1540	1680	1300	990	1620	1700	1460	2250	2345
12	2530	2860	2960	2380	1390	1580	1770	1080	1590	1380	310	120	0	720	900	1110	1500	1290	1390	1600	1660	1800	1420	1110	1740	1820	1580	2370	2465
13	2820	2250	2240	2970	1870	2060	2250	1560	2070	1860	790	600	720	0	180	390	780	570	670	880	940	1080	700	390	1020	1100	860	1650	1745
14	2640	2070	2060	2790	2050	2240	2430	1740	2250	2040	970	780	900	180	0	570	600	750	850	700	760	900	520	210	840	920	680	1470	1565
15	3210	2640	2630	3360	2260	2450	2640	1950	2460	2250	1180	990	1110	390	570	0	930	180	280	830	1330	1470	1090	780	1410	1490	1250	2040	2135
16	2680	2110	2100	2830	2650	2840	3030	2340	2850	2640	1570	1380	1500	780	600	930	0	750	650	100	800	940	560	390	880	960	720	1510	1605
17	3430	2760	2750	3540	2440	2630	2820	2130	2640	2430	1360	1170	1290	570	750	180	750	0	100	650	1510	1650	1270	960	1590	1670	1430	2220	2315
18	3330	2660	2650	3480	2540	2730	2920	2230	2740	2530	1460	1270	1390	670	850	280	650	100	0	550	1450	1590	1210	1040	1530	1610	1370	2160	2255
19	2780	2210	2200	2930	2750	2940	3130	2440	2950	2740	1670	1480	1600	880	700	830	100	650	550	0	900	1040	660	490	980	1060	820	1610	1705
20	2840	2270	2260	2990	2810	3000	3190	2500	3010	2800	1730	1540	1660	940	760	1330	800	1510	1450	900	0	140	720	550	240	260	80	1670	1765
21	2980	2410	2400	3130	2950	3140	3330	2640	3150	2940	1870	1680	1800	1080	900	1470	940	1650	1590	1040	140	0	860	690	200	120	220	1810	1905
22	2120	1550	1540	2270	2570	2760	2640	2260	2770	2560	1490	1300	1420	700	520	1090	560	1270	1210	660	720	860	0	310	800	880	640	950	1140
23	2430	1860	1850	2580	2260	2450	2640	1950	2460	2250	1180	990	1110	390	210	780	390	960	1040	490	550	690	310	0	630	710	470	1260	1355
24	2920	2350	2340	3070	2890	3080	3270	2580	3090	2880	1810	1620	1740	1020	840	1410	880	1590	1530	980	240	200	800	630	0	80	160	1750	1845
25	3000	2430	2420	3150	2970	3160	3350	2660	3170	2960	1890	1700	1820	1100	920	1490	960	1670	1610	1060	260	120	880	710	80	0	240	1830	1925
26	2760	2190	2180	2910	2730	2920	3110	2420	2930	2720	1650	1460	1580	860	680	1250	720	1430	1370	820	80	220	640	470	160	240	0	1590	1685
27	1170	600	590	1320	2070	2060	1690	2380	2070	2260	2440	2250	2370	1650	1470	2040	1510	2220	2160	1610	1670	1810	950	1260	1750	1830	1590	0	190

1320 2070 2060 1690 2380 2070 2260 2535 2345 2465 1745 1565 2135 1605 2315 2255 1705 1765 1905 1140 1355 1845

Table 1. Shortest distances among 28 attractions at Tokyo Disney Sea (m).

1925 1685

0

190

16 Caravan Carousel

20 Ariel's Playground

21 Jumpin's Jellyfish

22 Scuttle's Scooters

26 The Whirlpool

24 Blowfish Balloon Race

25 Mermaid Lagoon Theater

17 Jasmine's Flying Carpets

19 The Magic Lamp Theater

18 Sindbad's Storybook Voyage

23 Flounder's Flying Fish Coaster

27 20,000 Leagues Under the Sea

28 Journey to the Center of the Earth

<b>Table 2.</b> Playing time, queuing time, and satisfaction degree for each attraction in Tokyo Disney Sea.								
Attraction	Playing Time (min)	Satisfaction Degree	Queuing Time (min)					
1 Disney Sea Transit Steamer Line	7	0.1	10					
2 Fortress Explorations	8	0.3	10					
3 Venetian Gondolas	11.5	0.2	40					
4 Turtle Talk	30	0.4	35					
5 Tower of Terror	2	0.7	80					
6 Disney Sea Electric Railway	2.5	0.2	20					
7 Disney Sea Transit Steamer Line	13	0.2	20					
8 Toy Story Mania	5	0.3	100					
9 Big City vehicles	10	0.1	10					
10 Aquatopia	2.5	0.2	25					
11 Disney Sea Electric Railway	2.5	0.2	20					
12 Nemo & Friends Sea Rider	5	0.4	45					
13 Indiana Jones Adventure	3	0.8	80					
14 Disney Sea Transit Steamer Line	6	0.2	30					
15 Raging Spirits	1.5	0.7	110					

0.5

0.5

0.4

0.4

0.3

0.5

0.5

0.7

0.5

0.4

0.5

0.6

0.7

Table 2. Playing

2.5

1.5

10

23

10

1

1.5

1

1.5

14

1.5

5

3

The opening and the closing times for the theme park are 09:00 and 18:00. The weights of the objectives  $(c_1, c_2, c_3)$  are set to 0.2, 0.5, and 0.3, respectively, and conversion factors of utility  $(\lambda_1, \lambda_2, \lambda_3)$  are set to 1, 10, and 0.1, respectively. The population size is set to 200, and the maximum iteration is 200. The probabilities of the three genetic operators are generated randomly. This experiment was implemented using MATLAB.

The evolution result of the PGA is shown in the Figure 3, where the best solution in the experiment can be found when the iteration is 109, and the value of the fitness function is 36.0801.



Figure 3. Evolution results of PGA with 28 attractions.

12

15

15

25

10

20

20

35

30

30

30

30

130

The obtained satisfaction route is 0-2-27-22-24-21-20-26-23-16-19-18-17-13-11-6-0. The details of the results are illustrated in Table 3. The table contains seven items: recommended attraction, recommended arrival time, queuing time, time spent at the attraction, departure time from the attraction, walking time from this node to the next attraction, and the degree of satisfaction for selected attractions. The total walking time of the travel route is 92 min, the total queuing time is 372 min, the satisfaction of the tourists is 6.9, and the number of recommended attractions is 15.

Attraction	Arrival Time	Queuing Time (min)	Playing Time (min)	Departure Time	Walking Time to the Next Location (min)	Satisfaction Degree
Entrance (0)	09:00	-	-	9:00	11'15″	-
2	09:11	10	8	9:29	5'38	0.3
27	09:35	30	5	10:10	9'8″	0.6
22	10:19	20	1'30″	10:40	7'44″	0.5
24	10:48	30	1'30″	11:20	1'57"	0.5
21	11:22	20	1	11:43	1'22″	0.5
20	11:44	10	10	12:04	1	0.3
26	12:05	30	1'30″	12:36	4'39″	0.5
23	12:41	35	1	13:17	3'53″	0.7
16	13:21	12	2'30"	13:35	1	0.5
19	13:36	25	23	14:24	5'34″	0.4
18	14:29	15	10	14:54	1'2"	0.4
17	14:55	15	1'30″	15:11	5'49″	0.5
13	15:17	80	3	16:40	6'14''	0.8
11	16:46	20	2'30"	17:08	17'15″	0.2
6	17:25	20	2'30"	17:47	8'	0.2
Exit (0)	17:55	-	-	-	-	-

Table 3. Results of the satisfaction route with PGA (28 attractions).

Then, the evolution results of the improved APGA are illustrated in Figure 4. Here, we set the initial temperature to 90 and the cooling coefficient to 0.99. The best solution in the experiment can be found when the iteration is 156, and the value of fitness function is 36.0801. The result is the same as that of PGA, and the obtained satisfaction route is 0-2-27-22-24-21-20-26-23-16-19-18-17-13-11-6-0. The details of the results are illustrated in Table 4. As with Table 3, the table also contains seven items: recommended attraction, recommended arrival time, queuing time, playing time spent at the attraction, departure time from the attraction, walking time from this node to the next attraction, and the degree of satisfaction for selected attractions. The total walking time of the travel route is 92 min, the total queuing time is 372 min, the satisfaction of the tourists is 6.9, and the number of recommended attractions is also 15.



Figure 4. Evolution results of APGA with 28 attractions.

Attraction	Arrival Time	Queuing Time (min)	Playing Time (min)	Departure Time	Walking Time to the Next Location (min)	Satisfaction Degree
Entrance (0)	09:00	-	-	9:00	11'15″	-
2	09:11	10	8	9:29	5'38	0.3
27	09:35	30	5	10:10	9'8″	0.6
22	10:19	20	1'30″	10:40	7'44″	0.5
24	10:48	30	1'30″	11:20	1'57"	0.5
21	11:22	20	1	11:43	1'22″	0.5
20	11:44	10	10	12:04	1	0.3
26	12:05	30	1'30″	12:36	4'39″	0.5
23	12:41	35	1	13:17	3'53″	0.7
16	13:21	12	2 <b>'</b> 30″	13:35	1	0.5
19	13:36	25	23	14:24	5'34″	0.4
18	14:29	15	10	14:54	1'2"	0.4
17	14:55	15	1'30″	15:11	5'49″	0.5
13	15:17	80	3	16:40	6'14''	0.8
11	16:46	20	2'30"	17:08	17'15"	0.2
6	17:25	20	2'30"	17:47	8′	0.2
Exit (0)	17:55	-	-	-	-	-

Table 4. Results of the satisfaction route with APGA (28 attractions).

By comparing the two results above, when there are 28 attractions, the best solution of the two different algorithms can be found in a short time. In addition, we tested the two algorithms 15 times. For the same example, Table 5 includes the test results using the PGA 15 times, and the second row is the optimal value of fitness function in the experiment corresponding to each test, and the third row is the obtained satisfaction route. Table 6 illustrates the test result acquired using the APGA 15 times. From Tables 5 and 6, after being tested 15 times, the best solution was found once by using the PGA, while the best solution of 36.0801 was found twice by using the APGA. Table 7 illustrates the comparison results of the two different algorithms. The presented results are the best solution (Best), worst solution (Worst), average solution (Average), and the standard deviation (Std).

Number	The Optimal Value of Fitness Function	The Obtained Satisfaction Route				
1	35.5772	0-2-16-18-17-13-14-23-20-21-25-24-26-22-27-1-0				
2	35.611	0-6-13-17-18-19-16-20-21-24-26-23-22-27-2-1-0				
3	35.611	0-6-13-17-18-19-16-24-21-20-26-23-22-27-2-1-0				
4	35.9203	0-16-19-18-17-13-23-20-21-25-24-26-22-27-2-0				
5	35.1115	0-2-27-22-24-21-20-26-16-18-17-23-13-5-0				
6	35.611	0-6-13-17-18-19-16-24-21-20-26-23-22-27-2-1-0				
7	35.9202	0-20-21-25-24-26-16-19-18-17-13-23-22-27-2-0				
8	35.9216	0-2-16-19-18-17-13-23-20-21-25-24-26-22-27-0				
9	35.5742	0-6-17-18-16-20-21-25-24-26-13-23-22-27-2-1-0				
10	36.0801	0-2-27-22-24-21-20-26-23-16-19-18-17-13-11-6-0				
11	35.5434	0-6-12-13-17-18-16-24-21-20-26-23-22-27-2-1-0				
12	34.9208	0-16-19-18-17-13-23-21-25-24-26-22-27-2-1-0				
13	35.5964	0-6-13-17-18-16-20-21-25-24-26-23-22-27-2-1-0				
14	36.0655	0-6-11-13-17-18-16-20-21-25-24-26-23-22-27-2-0				
15	35.611	0-6-13-17-18-19-16-24-21-20-26-23-22-27-2-1-0				

Table 5. The results of 15 tests using the PGA (28 attractions).

Number	The Optimal Value of Fitness Function	The Obtained Satisfaction Route				
1	35.9216	0-2-24-25-21-20-26-16-19-18-17-13-23-22-27-0				
2	35.9202	0-20-21-25-24-26-16-19-18-17-13-23-22-27-2-0				
3	35.9203	0-16-19-18-17-13-23-20-21-25-24-26-22-27-2-0				
4	36.0655	0-6-11-13-17-18-16-20-21-25-24-26-23-22-27-2-0				
5	35.9216	0-2-24-25-21-20-26-16-19-18-17-13-23-22-27-0				
6	35.611	0-6-13-17-18-19-16-24-21-20-26-23-22-27-2-1-0				
7	35.7652	0-1-2-27-22-24-21-20-26-23-16-18-17-13-11-7-6-0				
8	36.08	0-6-11-13-17-18-19-16-26-24-21-20-23-22-27-2-0				
9	35.9199	0-26-20-21-25-24-16-19-18-17-13-23-22-27-2-0				
10	35.9203	0-16-19-18-17-13-23-20-21-25-24-26-22-27-2-0				
11	36.0801	0-6-11-13-17-18-19-16-24-21-20-26-23-22-27-2-0				
12	35.9203	0-16-19-18-17-13-23-20-21-25-24-26-22-27-2-0				
13	36.0801	0-2-27-22-24-21-20-26-23-16-19-18-17-13-11-6-0				
14	35.9216	0-2-24-25-21-20-26-16-19-18-17-13-23-22-27-0				
15	35.9202	0-20-21-25-24-26-16-19-18-17-13-23-22-27-2-0				

Table 6. The results of 15 tests using the APGA (28 attractions).

**Table 7.** Comparison of different methods for solving time-varying theme park routing (TDTPRP) (28 attractions).

Method	Best	Worst	Average	Std
PGA	36.0801	34.9208	35.645	0.31
APGA	36.0801	35.611	35.931	0.12

The test results indicate that both algorithms can be used to find the best solution within a short time when resolving the path of a theme park with 28 attractions. However, the APGA is superior to the PGA in terms of optimizing ability.

# 4.2. Randomly Generated Problem Instances

The results above demonstrate that both the PGA and the APGA mentioned in this paper can be used to find the best solution within a short time when resolving small and medium-sized theme park routing problems. Then, we expanded the test scale in the second test. We generated a theme park with one entrance and 60 attractions at random within a 2000  $\times$  2000 test environment, as illustrated in Figure 5.



Figure 5. Scatter plot

One of the points serves as the entrance (exit), and the remaining 60 points are regarded as the attractions. A visitor departing from the entrance of the theme park visits the attractions and then leaves the theme park. We randomly generated a simulated theme park environment and assumed that any two points are connected, and the distance between any two points is known. The time spent at the attraction, the queuing time, and the degree of satisfaction for each project are provided where the degree of satisfaction for the attractions are also generated from this interval [0, 1]. As with the previous experiment, we chose to use the same function to calculate the walking speed of the visitors.

The opening and the closing times are set as 09:00 and 17:00. The weights of the objectives  $(c_1, c_2, c_3)$  are set to 0.2, 0.5, and 0.3, respectively, and conversion factors of utility  $(\lambda_1, \lambda_2, \lambda_3)$  are set to 1, 10, and 0.1, respectively. The population size is set to 200, and the maximum iteration is 200. The probabilities of the three genetic operators are generated randomly.

The evolution results of the PGA are illustrated in Figure 6, where the best solution in the experiment can be found when the iteration is 185, and the best value of fitness function is 34.0835.



Figure 6. Evolution results of PGA with 60 attractions.

The obtained satisfaction route is 0-6-18-26-7-10-30-9-44-8-21-0. Details on the results are illustrated in Table 8. The total walking time of the travel route is 61 min, the total queuing time is 282 min, the degree of satisfaction of the tourists is 8.53, and the number of recommended attractions is 10.

Attraction	Arrival Time	Queuing Time (min)	Playing Time (min)	Departure Time	Walking Time to the Next Location (min)	Satisfaction Degree
Entrance (0)	09:00	-	-	9:00	10'55″	-
6	09:11	29	18	9:58	6	0.94
18	10:04	24	14	10:42	4'58"	0.92
26	10:47	38	11	11:36	6'20"	0.93
7	11:42	12	12	12:06	1'14"	0.68
10	12:08	29	15	12:52	6'40''	0.8
30	12:58	35	9	13:42	1′6″	0.91
9	13:44	15	17	14:16	1'8"	0.97
44	14:17	34	8	14:59	11'8″	0.79
8	15:10	31	17	15:58	10'3"	0.89
21	16:08	35	16	16:59	1′	0.7
Exit (0)	17:00	-	-	-	-	-

Table 8. Results of the satisfaction route with PGA (60 attractions)

The evolution result of the improved APGA is illustrated in Figure 7, where the best solution in the experiment can be found when the iteration is 160, and the best value of fitness function is 34.1492. The obtained satisfaction route is 0-6-18-26-24-7-10-30-9-44-8-0. The details of the results are illustrated in Table 9. The total walking time of the travel route is 65 min, the total queuing time is 272 min, the satisfaction of tourists is 8.51, and the number of recommended attractions is 10.



Figure 7. Evolution results of APGA with 60 attractions.

Attraction	Arrival Time	Queuing Time (min)	Playing Time (min)	Departure Time	Walking Time to the Next Location (min)	Satisfaction Degree
Entrance (0)	09:00	-	-	9:00	10'55″	-
6	09:11	29	18	9:58	6	0.94
18	10:04	24	14	10:42	4'58"	0.92
26	10:47	38	11	11:36	5'40"	0.93
24	11:42	25	12	12:19	5'23"	0.68
7	12:24	12	12	12:48	1'15″	0.68
10	12:50	29	15	13:34	6'43"	0.8
30	13:40	35	9	14:24	1′6″	0.91
9	14:25	15	17	14:57	1'33″	0.97
44	14:59	34	8	15:41	11'14″	0.79
8	15:52	31	17	16:40	9'53″	0.89
Exit (0)	16:50	-	-	-	-	-

Table 9. Results of the satisfaction route with APGA (60 attractions)

If there are 28 attraction projects, the two different algorithms can be used to find the relative best solution within a short time successfully. However, when we increased the attractions to 60, the computation time of the PGA was longer, and it was inferior to the APGA in terms of optimizing ability. By comparing Figures 6 and 7, the PGA was stuck in a locally optimal solution in the 185th generation, the best result was 34.0835, and it failed to reach the ideal result. By contrast, the APGA overcame the weakness of being stuck in local optimization and tended to be stable in the 160th generation. The best solution in this experiment was 34.1492.

Similarly, we ran the two algorithms 15 times in the test. Table 10 illustrates the test results acquired by using the PGA 15 times. Table 11 illustrates the test results acquired by the APGA. It can be concluded from the results that, if the PGA was used for the solution, the positive solution rate was 0, which means no best solution was found in the solution process using 15 tests. The best solution was found twice once the APGA was used, and Table 12 illustrates the comparison of the two algorithms.

Number	The Optimal Value of Fitness Function	The Obtained Satisfaction Route
1	32.5523	0-6-24-18-26-49-7-9-30-8-54-0
2	34.0835	0-6-18-26-7-10-30-9-44-8-21-0
3	34.0067	0-10-7-24-6-18-26-30-9-44-8-0
4	31.3534	0-6-18-26-7-30-9-8-34-54-0
5	32.1111	0-6-18-26-7-10-30-9-8-13-54-0
6	33.9866	0-6-18-26-7-10-44-30-9-8-42-0
7	32.2745	0-13-6-18-26-30-9-7-10-8-54-0
8	33.8962	0-24-7-10-44-9-30-26-18-53-6-0
9	32.1077	0-8-30-9-10-26-18-53-6-7-13-0
10	33.9753	0-24-26-18-6-7-10-30-9-44-8-0
11	32.4752	0-24-6-18-26-49-7-30-9-8-54-0
12	33.5624	0-42-8-9-30-10-7-26-18-6-21-0
13	32.4777	0-6-53-18-26-9-30-7-10-13-8-0
14	32.5079	0-8-13-10-7-9-30-26-18-53-6-0
15	33.2479	0-42-10-7-9-30-26-18-53-6-24-0

Table 10. The results of 15 tests using the PGA (60 attractions)

Number	The Optimal Value of Fitness Function	The Obtained Satisfaction Route
1	34.1265	0-8-44-9-30-10-7-26-18-24-6-0
2	34.0955	0-6-18-26-24-7-10-44-9-30-8-0
3	34.0302	0-8-10-7-9-44-30-26-18-24-6-0
4	34.0905	0-8-44-9-30-10-7-26-18-6-21-0
5	34.1107	0-8-44-9-30-10-7-26-18-6-24-0
6	34.0365	0-8-44-9-30-10-7-26-18-24-6-0
7	34.1492	0-6-18-26-24-7-10-30-9-44-8-0
8	34.0681	0-6-24-18-26-10-7-9-30-44-8-0
9	34.1492	0-6-18-26-24-7-10-30-9-44-8-0
10	34.0142	0-6-18-26-10-7-30-9-44-8-42-0
11	34.1399	0-6-18-26-24-10-7-30-9-44-8-0
12	33.9915	0-6-18-26-7-10-44-9-30-8-42-0
13	34.0815	0-6-24-18-26-30-9-44-7-10-8-0
14	34.0679	0-24-6-18-26-30-9-44-7-10-8-0
15	34.1307	0-6-24-18-26-7-10-30-9-44-8-0

Table 11. The results of 15 tests using the APGA (60 attractions)

Table 12. Comparison of different methods for solving TDTPRP (60 attractions)

Method	Best	Worst	Average	Std
PGA	34.0835	31.3534	32.9746	0.86
APGA	34.1492	33.9915	34.0854	0.05

# 5. Discussion

In order to solve the problem of crowded queuing in large theme parks, improve visitor satisfaction, and reduce congestion, a time-dependent theme park routing problem (TDTPRP) was proposed to maximize the utility of the visitors and minimize queuing and walking times for selecting the optimal attractions under the framework of the traveling salesman problem (TSP), where walking time was treated as time-dependent and changed according to different time periods. This model can provide a more accurate plan and plan for a single decision. To solve the proposed model and verify the feasibility and the effectiveness of the model, we used a parthenogenetic algorithm and proposed an annealing parthenogenetic algorithm. In the experimental stage, we conducted two experiments; the experimental data were divided into real-world problems and randomly generated problems. First, we used the Tokyo Disney Sea with 28 attractions as the real-world problem to prove the effectiveness of the model and the algorithm. Second, to prove the stability of the model and the algorithm, we expanded the scale of the experiment and randomly generated an example of a theme park with 60 attractions within a  $2000 \times 2000$  test environment. The results demonstrate that, when the experiment scale is small, the general parthenogenetic algorithm and the annealing parthenogenetic algorithm have the same excellent optimization ability, but the annealing parthenogenetic algorithm has better optimization ability than the general parthenogenetic algorithm when the data scale was expanded.

For future research, there is space for further improvements. For example, the walking time is dynamic, but to make it comparable to a realistic problem, the queuing time should also be constantly changing. Moreover, at present, some theme parks offer tickets to skip queues, some of which are free and some of which involve a fee. These factors could be considered in future research to continue to develop the model, and with the improvement of the model, the design of the algorithm should also continue to improve.

**Author Contributions:** Conceptualization, Z.Y.; methodology, Z.Y. and J.L.; software, Z.Y. and J.L.; validation, Z.Y., J.L. and L.L.; formal analysis, Z.Y.; investigation, Z.Y.; resources, Z.Y.; data curation, Z.Y.; writing-original draft preparation, Z.Y.; writing-review and editing, J.L. and L.L.; supervision, L.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

## References

- Johns, N.; Gyimóthy, S. Mythologies of a theme park: An icon of modern family life. *J. Vacat. Mark.* 2002, 8, 320–331. [CrossRef]
- Kawamura, H.; Kurumatani, K.; Ohuchi, A. A study on coordination scheduling algorithm for theme park problem with multiagent. *Inst. Electron. Inform. Commun. Eng. Tech. Rep. Artif. Intell. Knowl.-Based Process.* 2003, 102, 25–30.
- 3. Yasushi, Y.; Keiji, S. *Comparison of Traversal Strategies in Theme Parks;* Technical Report for IPSJ SIG Technical Reports (ICS-147); IPSJ: Tokyo, Japan, 2007; pp. 15–22.
- 4. Kikuchi, S.; Onishi, K. A Sightseeing Route Guidance System using Java Applet. In Proceedings of the 54th National Convention IPSJ 1, Chiba, Japan, 12–14 March 1997; pp. 431–432.
- 5. Matsuda, Y.; Nakamura, M.; Kang, D.; Miyagi, H. An Optimal Routing Problem for Sightseeing and its Solution. *IEEJ Trans. Electron. Inf. Syst.* **2004**, *124*, 1507–1514. [CrossRef]
- 6. Tsai, C.-Y.; Chung, S.-H. A personalized route recommendation service for theme parks using RFID information and tourist behavior. *Decis. Support Syst.* **2012**, *52*, 514–527. [CrossRef]
- 7. Lee, C.-S.; Chang, Y.-C.; Wang, M.-H. Ontological recommendation multi-agent for Tainan City travel. *Expert Syst. Appl.* **2009**, *36*, 6740–6753. [CrossRef]
- 8. Lim, K.H.; Chan, J.; Leckie, C.; Karunasekera, S. Personalized trip recommendation for tourists based on user interests, points of interest visit durations and visit recency. *Knowl. Inf. Syst.* **2017**, *54*, 375–406. [CrossRef]
- Mor, M.; Sagi, D. Computing touristic walking routes using geotagged photographs from Flickr. In Proceedings of the 14th International Conference on Location Based Services, Zurich, Switzerland, 15–17 January 2018.
- Gionis, A.; Lappas, T.; Pelechrinis, K.; Terzi, E. Customized tour recommendations in urban areas. In Proceedings of the 7th ACM International Conference on Web Search and Data Mining—WSDM '14, New York, NY, USA, 24–28 February 2014; Association for Computing Machinery (ACM): New York, NY, USA, 2014; pp. 313–322.
- 11. Sengupta, L.; Mariescu-Istodor, R.; Fränti, P. Planning Your Route: Where to Start? *Comput. Brain Behav.* **2018**, *1*, 252–265. [CrossRef]
- 12. Niitsuma, H.; Arai, K.; Ohta, M. Efficient Constraints for Tour Recommendation. *Inform. Process. Soc. Jpn. Database (TOD)* **2016**, *9*, 34–45.
- 13. Matsuda, Y.; Nakamura, M.; Kang, D.; Miyagi, H. An optimal routing problem for sightseeing with fuzzy time-varying weights. In Proceedings of the 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583), The Hague, The Netherlands, 10–13 October 2004; Volume 4.
- 14. Little, J.D.C.; Murty, K.G.; Sweeney, D.W.; Karel, C. An Algorithm for the Traveling Salesman Problem. *Oper. Res.* **1963**, *11*, 972–989. [CrossRef]
- 15. Laporte, G.; Martello, S. The selective travelling salesman problem. *Discret. Appl. Math.* **1990**, 26, 193–207. [CrossRef]
- 16. Feillet, D.; Dejax, P.; Gendreau, M. Traveling Salesman Problems with Profits. *Transp. Sci.* 2005, 39, 188–205. [CrossRef]
- 17. Mak-Hau, V.; Thomadsen, T. Polyhedral combinatorics of the cardinality constrained quadratic knapsack problem and the quadratic selective travelling salesman problem. *J. Comb. Optim.* **2006**, *11*, 421–434. [CrossRef]
- 18. Vansteenwegen, P.; Souffriau, W.; Van Oudheusden, D. The orienteering problem: A survey. *Eur. J. Oper. Res.* **2011**, *209*, 1–10. [CrossRef]
- 19. Ergezer, H.; Leblebicioglu, K. Path Planning for UAVs for Maximum Information Collection. *IEEE Trans. Aerosp. Electron. Syst.* **2013**, *49*, 502–520. [CrossRef]
- 20. Gendreau, M.; Laporte, G.; Semet, F. A tabu search heuristic for the undirected selective travelling salesman problem. *Eur. J. Oper. Res.* **1998**, *106*, 539–545. [CrossRef]
- 21. Qin, H.; Lim, A.; Xu, D. *The Selective Traveling Salesman Problem with Regular Working Time Windows;* Springer Science and Business Media LLC: Berlin/Heidelberg, Germany, 2009; pp. 291–296.
- 22. Ke, L.; Archetti, C.; Feng, Z. Ants can solve the team orienteering problem. *Comput. Ind. Eng.* 2008, 54, 648–665. [CrossRef]

- 23. Bouzarth, E.L.; Forrester, R.J.; Hutson, K.R.; Isaac, R.; Midkiff, J.; Rivers, D.; Testa, L. A Comparison of Algorithms for Finding an Efficient Theme Park Tour. *J. Appl. Math.* **2018**, 2018, 1–14. [CrossRef]
- 24. Kanoh, H.; Ochiai, J. Solving Time-Dependent Traveling Salesman Problems Using Ant Colony Optimization Based on Predicted Traffic. In *Advances in Soft Computing*; Springer Science and Business Media LLC: Berlin/Heidelberg, Germany, 2012; Volume 151, pp. 25–32.
- 25. Yanfeng, L.; Jun, L.; Da, Z. Dynasearch Algorithms for Solving Time Dependent Traveling Salesman Problem. *J. SouthWest JiaoTong Univ.* **2008**, *43*, 187–193.
- Maojun, L.; Tong, T. A parthenogenetic algorithm and analysis on its global convergence. *Acta Autom. Sin.* 1999, 25, 68–72.
- 27. Gandomkar, M.; Vakilian, M.; Ehsan, M. A combination of genetic algorithm and simulated annealing for optimal DG allocation in distribution networks. In Proceedings of the Canada Conference on Electrical and Computer Engineering, Saskatoon, SK, Canada, 1–4 May 2005. [CrossRef]
- 28. Tian, Q.C.; Pan, Q.; Wang, F.; Zhang, H.C. Research on Learning Algorithm of BP Neural Network Based on the Metropolis Criterion. *Tech. Autom. Appl.* **2003**, *5*, 15–17.
- 29. Liu, L.; Olszewski, P.; Goh, P.-C. Combined Simulated Annealing and Genetic Algorithm Approach to Bus Network Design. In *International Conference on Transport Systems Telematics*; Springer Science and Business Media LLC: Berlin/Heidelberg, Germany, 2010; pp. 335–346.
- 30. Suppapitnarm, A.; Seffen, K.A.; Parks, G.T.; Clarkson, P.J. A simulated annealing algorithm for multiobjective optimization. *Eng. Optim.* 2000, *33*, 59–85. [CrossRef]

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).