For office use only
T1 ________________
T2 ________________
T3 ________________
T4 ________________

For office use only
F1 ________________
F2 ________________
F3 ________________
F4 ________________

2016

19th Annual High School Mathematical Contest in Modeling (HiMCM) Summary Sheet

(Please make this the first page of your electronic Solution Paper.)

Team Control Number: 6453

Problem Chosen: A

Summary

In order to achieve the goal of reducing road congestion and ensuring participants proceed without hindrance, at the same time solve the problem of minimizing the competition time, this paper adopts the following 3 models to obtain solution:

First of all, in model1, according to the analysis of data from the last year competition, we extract the mean and standard deviation as basis of classification. We apply Cluster Analysis to divide all the participants into 6 groups, combining several original groupings and let the athletes at the similar level to be on the same group; the strengths of doing this are: reduce the total time of competition (avoid the time issue due to the overmuch division originally given by the question), ensure the participants will not cause hindrance to each other (competitors with similar level will not cause hindrance), and reduce the problem of road congestion.

In model 2, we apply the Genetic Algorithm to optimize the starting order and exact starting time of every group. We make the total time to be smallest under the condition of athletes in every group will not disturb mutually, no road congestion. The result turns out to be: the total competition time can be controlled within 4.5 hours, which achieves the goal of organizer. This algorithm can be applied to find the optimum solution in the shortest amount of time. This model can be used in common scheduling problem because of strong representation.

In model 3, we apply linear programming. First, we use variable-controlling method, under the result of optimum starting sequence of athletes derived in model 2; we consider the influence that distances of races will exert on road congestion and total competition time. For only minimizing the distances of race to optimize the total time is meaningless, this paper assumes the constricted condition of race distances and finally derives the linear programming model. The result shows, adjusted distance can farthest reduce the waiting time, which is set for the purpose of avoiding road congestion and mutual disturbance; as a result, maximize the efficiency of using time.

In conclusion, this paper solves and gives complete solution of the problem of reducing road congestion and minimizing total competition time, successfully achieve the goal of question. This paper has strong generality therefore can be applied to daily scheduling problem.

Two-page letter to the Mayor

Team6453
Nov, 2016

Mayor
Town,
U.S.A

Dear Mayor,

It's honorable for us to participate in planning for this year's triathlon competition. We are glad to have this chance working with you and organize the triathlon to support our youth organization.

We reorganize the groupings compared to last year. For the consideration of maintaining a variety of divisions at the same time reducing road closure time, we combine the groups with competitors of similar level and suitable numbers of people to be at the same category, and finally derive 6 groups. Due to the big scale of several groups, we separate those groups to let them start at different time.

The starting sequence of every group is made according to the optimum solution we derived from model, which reduces the road congestion and minimizes the road closure time. The specific starting schedule is attached on the next page. The total time of competition is controlled within 4.5 hours, which achieves the goal.

Thank you again for giving us the chance to take part in preparation, and wish the competition a great success!

Sincerely yours,
Team 6453

Race Day Event Schedule

Groups		Start time/mins	number
M_PRE &F_PRO&M_PRO	M_PRE &F_PRO&M_PRO[1]	00:00:00	20
	M_PRE &F_PRO&M_PRO[2]	00:00:01	19
F_OPEN	F OPEN[1]	00:01:00	20
	F OPEN[2]	00:01:01	20
	……	……	……
	F_OPEN[28]	00:01:28	18
ATH	ATH[1]	00:02:00	20
CLY	CLY[1]	00:38:50	20
	CLY[2]	00:38:51	17
M OPEN	M OPEN[1]	01:02:10	20
	M OPEN[2]	01:02:11	20
	……	……	……
	M_OPEN[67]	01:03:16	15
F PREMIER	F PREMIER[1]	01:58:40	11

Content

Swim, Bike, and Run

I. Introduction

Triathlon is a competition consisting of 3 stages, a scheduling plan is crucial to a big-scale triathlon competition. There are diverse participants that are very different in level, so we need reasonable grouping to minimize total time. There are too many groups, so we have to make starting sequence to reduce road congestion and optimize control time.

We need to give plan to solve the questions below:

(1) Attract participants with diverse divisions.

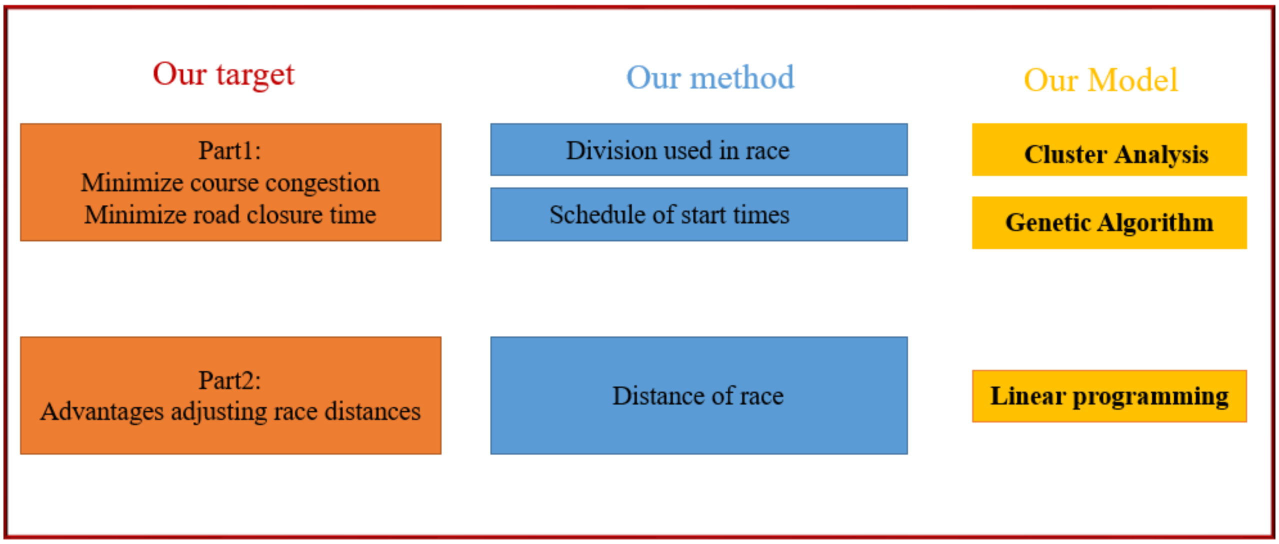
(2) Reduce road congestion and avoid mutual hindrance between participants.

(3) Minimize total competition time, specifically less than 5.5 hours.

Without implementing the model and analysis, the triathlon will not be successfully held, for the large amount of participants, and the long road closure time due to congestion.

After mathematically analyzing the above problems, our group give our solutions based on the results of modeling.

The structure of our model is as follows:



II. Basic Assumptions

1. The athletes at the similar level will not disturb each other.
2. There is no accident happening in the competition that forces the competition to end.
3. Every group will not have extra waiting time besides the converting time after setting off.

4. The athletes do uniform motion in every competition.
5. The velocity of athletes for every races will not change due to the change of distance set for race.

III. Definitions and Notations

Symbols	Definition
μ	The average value of the sample
σ	The standard deviation of the sample
a(i)	The sample value
$T_{\mathrm{s}}(j_{\mathrm{i}},\mathrm{k})$	Represents the time that group j_{i} starts to do the race k
$T_{\mathrm{e}}(j_{\mathrm{i}},\mathrm{k})$	Represents the time that group j_{i} finishes the race k
$T\ (j_{\mathrm{i}},\mathrm{k})$	Represents the time interval for group j_{i} during the race k.
$c_{\max}$	The maximum time it takes for all people to finish the whole Triathlon.
$\max T_{\mathrm{wait}}$	The maximum wait time for group j_{n} before the first race.
R	The race event velocity in traditional Olympic triathlon
T	Race event time in traditional Olympic triathlon
S	Race event distance in traditional Olympic triathlon
T_{ij}	The time that group i spends in the race j
c	The total transition time
S_{j}	The distance of race event j
T_{itotal}	The total time of group i in the whole triathlon

IV. Model 1: Ultimate Grouping Based on Cluster Analysis

1.1 Introduction to the model

We analyze the data according to the information about the results of athletes given by the question stem; we first classify athletes on the basis of the initial category and ages. Then further group the competitors of similar level together through cluster analysis, according to the statistic feature of the initial classification results.

This helps us to achieve efficient use of time, minimize course congestion; at the same time, we achieve the goal of having a variety of groups attracting athletes. The steps are as following pages.

1.2 Data preprocessing

First, we classified data according to the given categories (M Pro, F Pro, M Premier, F Premier, M Open, F Open, CLY, ATH) and ages. Specifically speaking for age, Because of the majority of competitors are in the group of OPEN, based on research, we set the basis of classification on older than 40 years old and younger than 40.

Assume the level of every group of athletes is in normal distribution. The two key parameters in the model are expected value μ (mean), and sigma σ.

μ is the location parameter, and describes the central tendency of normal distribution; it represents the average group level and indicates the time needed averagely for a specific group to finish the race.

The calculating formula is:

$$\mu = \frac{a(0)+a(1)+\cdots+a(n-1)}{n} \tag{1.1}$$

where a(i) represents the sample value, n refers to number.

We calculate sigma σ of each group of data to achieve the goal of taking most of the competitors into account. σ describes the dispersion degree of normal distribution, therefore most of the data can be represented through the use of σ. Specifically in this case, we take nearly the slowest athletes into consideration to ensure the time will not exceed the regulation at the same time satisfy the prerequisite that no qualification time set up for athletes.

$$\sigma^2 = \frac{1}{n-1} \times [(a(0)-\mu)^2 + (a(1)-\mu)^2 + \cdots + (a(n-1)-\mu)]^2 \tag{1.2}$$

In order to simplify analyzing process, we converted the unit into minutes. The result presents as following：

Table 1 Preliminary classification and data processing result

CATEGORY		SWIM	T1	BIKE	T2	RUN	FINALTM	NUMBER
M PREMIER	μ	16.67	4.40	68.63	1.26	39.75	130.70	49
	σ	1.43	0.56	4.71	0.32	3.53	8.63	
M PRO	μ	13.14	3.27	62.17	0.86	35.33	114.77	7
	σ	1.22	0.34	2.92	0.09	1.86	5.10	
M OPEN1	μ	24.26	7.20	84.38	2.84	58.03	176.71	1101
	σ	4.22	2.91	12.07	1.38	10.75	26.27	
M OPEN2	μ	22.37	8.07	85.35	3.19	60.70	179.67	1046
	σ	4.12	3.04	11.20	1.62	12.26	26.54	
CLY	μ	25.63	8.04	88.27	3.47	71.35	196.76	60
	σ	3.83	3.20	12.88	2.13	13.73	30.04	
F OPEN1	μ	22.34	9.87	98.77	3.60	60.82	195.41	542
	σ	3.35	3.33	15.36	1.97	10.76	30.07	
F OPEN2	μ	22.71	10.27	97.57	3.85	64.71	199.10	356
	σ	3.37	3.58	14.13	2.14	12.75	31.29	
F PRO	μ	12.97	3.64	67.63	0.91	38.31	123.46	6
	σ	0.36	0.17	2.43	0.15	1.98	4.45	
F PREMIER	μ	17.03	4.95	76.27	1.44	44.13	143.82	18
	σ	1.40	0.63	4.60	0.48	3.75	8.40	
ATH_	μ	23.42	11.63	108.98	4.51	77.43	225.97	32
	σ	3.56	3.65	19.36	1.87	13.95	38.07	

1.3 Cluster Analysis

Cluster analysis or clustering is the task of grouping a set of objects that objects in the same group are more similar to each other than to those in other groups. Cluster analysis helps people get the relationship between variables more scientifically and accurately.

The original classification is not suitable for data analyzing because of the quantity is large. So, we use Cluster analysis to further classify. Assume the competitors on the same level need similar time to accomplish the race, therefore people within the group will not hinder mutually. Specifically, we do this classification according to two standards: (1) athletes with similar levels will be put in a group. (2) Numbers of people are suitable for grouping.

Furthermore, due to the similar required time, the grouping of same level competitors helps to reduce the road congestion. Suggest there are two people A, B having the same velocity v running on a playground. A set off at t1, and B set off at t1+Δt, they will never meet and cause congestion: there is always a distance of $\mathbf{v} \times \Delta\mathbf{t}$.

We put the results of preprocessing to the mathematical statistics software SPSS. On the basis of features of data from different groups, the result of cluster analysis as following:

Agglomeration Schedule

Stage	Cluster Combined		Coefficients	Stage Cluster First Appears		Next Stage
	Cluster 1	Cluster 2		Cluster 1	Cluster 2	
1	3	4	15.623	0	0	6
2	6	7	22.458	0	0	7
3	1	8	26.367	0	0	4
4	1	2	60.561	3	0	5
5	1	9	171.835	4	0	9
6	3	5	171.848	1	0	7
7	3	6	210.356	6	2	8
8	3	10	651.216	7	0	9
9	1	3	1819.957	5	8	0

Figure 1 Cluster Analysis result 1

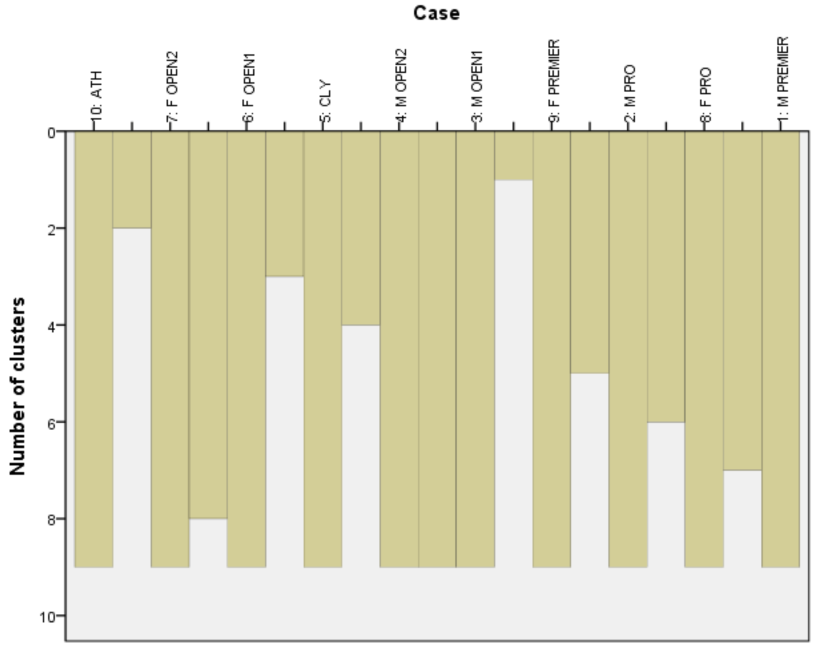


Figure 2 Cluster Analysis result 2

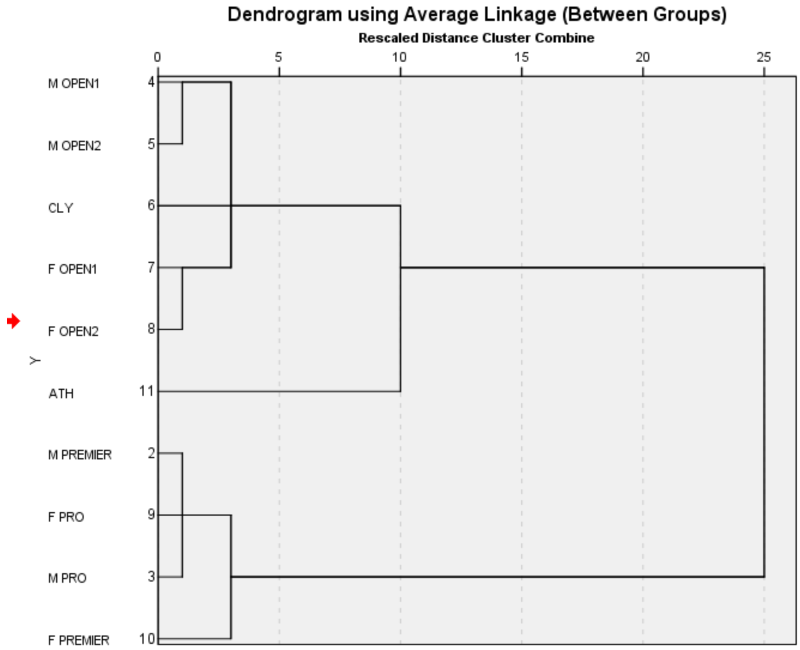


Figure 3 Cluster Analysis result 3

From the result of Cluster Analysis, we can see, although we classified M_OPEN and F_OPEN into two groups according to ages. But due to the similarity between the features of two statistic data, and the level of two groups are similar, we can combine this two groups into one. Ultimately, we divide all the athletes into 6 categories, the result is shown as following:

Table 2 Cluster Analysis result

CATEGORY		SWIM	T1	BIKE	T2	RUN	FINALTM	NUMBER
M OPEN	μ	23.34	7.63	84.85	3.01	59.33	178.15	2147
	σ	4.28	3.00	11.66	1.51	11.58	26.44	
F OPEN	μ	17.03	4.95	76.27	1.44	44.13	143.82	898
	σ	1.40	0.63	4.60	0.48	3.75	8.40	
ATH	μ	23.42	11.63	108.98	4.51	77.43	225.97	32
	σ	3.56	3.65	19.36	1.87	13.95	38.07	
M_PRE &F_PRO&M_PRO	μ	15.91	4.20	67.80	1.18	39.11	128.20	62
	σ	1.99	0.65	4.79	0.33	3.53	9.52	
F PREMIER	μ	17.03	4.95	76.27	1.44	44.13	143.82	18
	σ	1.40	0.63	4.60	0.48	3.75	8.40	
CLY	μ	25.63	8.04	88.27	3.47	71.35	196.76	60
	σ	3.83	3.20	12.88	2.13	13.73	30.04	

Because of the classification stated above is according to the data of last year. According to the question stem, the total number of athletes is 2000. In order to derive the final grouping under this case specifically, we predict the number of people of different category, formula is as following:

$$N_{predicted} = \frac{N_{provided}}{N_{provided\ total}} \times 2000 \tag{1.3}$$

Result shows as follow:

Table 3 Final predicted data for the triathlon

CATEGORY		SWIM	T1	BIKE	T2	RUN	FINALTM	NUMBER
M OPEN	μ	23.34	7.63	84.85	3.01	59.33	178.15	1335
	σ	4.28	3.00	11.66	1.51	11.58	26.44	
F OPEN	μ	17.03	4.95	76.27	1.44	44.13	143.82	558
	σ	1.40	0.63	4.60	0.48	3.75	8.40	
ATH	μ	23.42	11.63	108.98	4.51	77.43	225.97	20
	σ	3.56	3.65	19.36	1.87	13.95	38.07	
M_PRE &F_PRO&M_PRO	μ	15.91	4.20	67.80	1.18	39.11	128.20	39
	σ	1.99	0.65	4.79	0.33	3.53	9.52	
F PREMIER	μ	17.03	4.95	76.27	1.44	44.13	143.82	11
	σ	1.40	0.63	4.60	0.48	3.75	8.40	
CLY	μ	25.63	8.04	88.27	3.47	71.35	196.76	37
	σ	3.83	3.20	12.88	2.13	13.73	30.04	

The classification helps us to achieve efficient use of time, minimize course congestion; at the same time, we achieve the goal of having a variety of groups attracting athletes.

V. Model 2: the Triathlon race time model

2.1 The Description of the Triathlon race time model

The question requires to have 2000 runners and plan to have a traditional open Olympic triathlon, which consists of a 1500m swim, a 40K bike ride and a 10K run. The race will have variety of participants containing professional triathletes (best in the nation and world) and some premier triathletes (not quite professionals, but among the fastest amateurs in previous triathlons) and an open triathlon. Also, the CEO of the race sponsor wants to make sure that participants should be able to proceed without hindrance during each phase of the triathlon. According to the chart given we could conclude the question should satisfy these following hypotheses:

1). The sequence of three types of sports in this Triathlon race is fixed. That is, participants must do swimming first, then biking and finally running.

2). At the same time, we could have different groups of people doing different types of the three sports. For example, while one group of people is swimming, another early-started group could be riding as long as they do not interfere with each other.

3). People could only do one type of sport at the same time.

4). We consider the athletes' time in the transition area T1 and T2 into the counting of the total time.

5). Most of the athletes in the same group are at similar level, so they will not affect each other during the races.

2.2 The modeling for the Triathlon race model

According to the description above, we could convert it into a form that $T_s(j_i, \mathrm{k})$ refers to the starting time for group j_i to finish the k of the three races. Regard $T_e(j_i, \mathrm{k})$ as the time that group j_i finish the race *k*. $T(j_i, \mathrm{k})$ represents the time interval of the j_i group.

$\{j_1, j_2, \ldots\ldots, j_n\}$ represents the starting order of each group.

The max time it takes to finish the full race will be:

$$c_{\max} = T_e(j_n, m) \tag{2.1}$$

Thus, we could find the final group's finish time will be:

$$\min z = \min c_{\max} = \min T_e(j_n, m) \tag{2.2}$$

We could calculate the group *j*'s finishing time by adding the beginning time of *k* and the time interval for doing the race *k*:

$$T_e(j_i,k)=T_s(j_i,k-1)+T(j_i,k) \tag{2.3}$$

Similarly, when k>1, the time when j_i begins the *k*'s race is the sum of the *(k-1)*'s beginning time, the *(k-1)*'s finishing time and the transition time:

$$T_e(j_i,k)=T_s(j_i,k-1)+T(j_i,k-1)+T_{trans} \tag{2.4}$$

When $j_i>1$, j_i's starting time of the first race can be represented as:

$$T_s(j_i,1)=T_s(j_i-1,1)+\max T_{wait} \tag{2.5}$$

The math modeling of this problem is similar to the Flow Shop Scheduling, which has 3 machines and taking transformation time into consideration. As we know, it is a NP hard problem. In our paper, we use Genetic Algorithm to solve this problem.

2.3 Introduction of Genetic Algorithm

Genetic Algorithm(GA) is a meta-heuristic inspired by the process of natural selection of Darwin, specially the principle of "survival of the fittest". The Genetic Algorithm is invented by John Holland in the early 1970's to mimic some of the processes observed in natural revolution. Nowadays, GA could represent an intelligent exploitation of a random search used to solve optimization problems in various fields, like science, engineering and business.

The Genetic Algorithm could be formalized as GA= （$P(0),N,l,s,g,p,f,t$）, and $P(0)=(P_1(0),P_2(0)\ldots\ldots P_n(0))$ represents the initial population. *N* is the number that the population contains and *l* is the length of the binary string. *s* represents the selection strategy. *g* represents the genetic operator, which includes selection Q_r, crossover Q_c and mutation Q_m. *P* refers to the operation probability of the genetic operator. It contains P_r, P_c and P_m. *t* is the termination criterion.

Genetic Algorithm is helpful when solving complex problems because it contains these basic features of 1) intelligence, it has the ability of self-organizing, adaptivity and self-learning. 2)parallelism, the data manage and information searching both have the ability of parallelism. 3)multiplicity, we could get multiple approximate solutions by doing the Genetic Algorithm. 4)indeterminacy. Mutation operation may lead to the indeterminacy of the answer. 5) non directional.

It decreases the possibility to be caught in one part of the question but searching for answer through the strings of the population.

2.4 Composition and solving of the Genetic Algorithm

2.4.1 The composition of the Genetic Algorithm

The use of Genetic Algorithm contains 6 components: coding design, the setting of the initial population, genetic manipulation (including selection, crossover and mutation, etc.), algorithm control parameters setting and constraint conditions of processing.

1). the coding design

In the genetic algorithm, the method to describe the feasible solution of the problem, and transfer the solution of a problem from its solution space to the genetic algorithm can be identified as coding. The conversion from solution space that can be solved by genetic algorithm to the space of feasible solution of the problem is called decoding. Encoding methods contain: binary encoding, gray code encoding, floating-point encoding and real number encoding and so on.

In this question for the starting order of the race for each group, the commonly used method is to use the chromosome to represent the order of the team, for example, for the six groups' start time, the k's chromosome $v_k = [1,2,3,4,5,6]$ .It represent the starting time of the six groups： $j_1 \to j_2 \to j_3 \to j_4 \to j_5 \to j_6$.

2). the setting of the initial population

Genetic algorithm follows the rule of the "survival of the fittest" of nature, and it does not use the external information in the search of the evolution, but only use the degree of fitness function as a standard to judge the individuals. Generally speaking, the fitness function is transformed from the objective function. Therefore, the choice of fitness function directly affects the algorithm. The goal of the model is to make the total time of the game the shortest, that is, $\min z = \min c_{\max} = \min T_e(j_n, m)$. the higher the degree of adaptability of the genetic algorithm, the better the individual. We could use

$f(v_k) = \frac{1}{T_{\mathrm{emax}}^k}$ as our fitness function. Thus, when the maximum finishing time gets bigger, the fitness gets less good. Also, when the maximum finishing time gets smaller, the fitness is better.

(3) the generation of initial population

Genetic algorithm can be used in many ways to generate initial population. The use of different methods to generate the initial population will have a certain difference, the efficiency of the optimal solution will be different. Therefore, for the actual situation of the problem, to choose a suitable initial population generation method is advantageous to obtain the optimal solution as soon as possible. This question is a simple arrangement for all the athletes' race time, and the groups number is not large. Thus, we choose to use the method of random generation of initial population for the question.

(4) choosing

The purpose of selection is to select the strong individual in the group, so that they have the opportunity to generate new groups as the parent generation process. To judge whether an individual is superior, we should compare the fitness of each individual and those which has high individual fitness has higher probability to be selected, those have low individual fitness have a low probability that the individual will being inherited to the next generation. The selection strategy has nothing to do with the coding method. The common selection operator includes: the choice of fitness ratio, the selection of the best individual and so on. In this question, we choose to divide the athletes into groups based on their sex, skill and age. Thus, compare themselves and the one has lager fitness value could remain, and the one with lower fitness value will be eliminated.

(5) cross

Cross, also known as the reorganization, is use the larger probability to select two individuals from the group, and exchange one or some of the two individuals. The crossover operator generates the sub generation, which will inherit the basic characteristics of the father. The design of crossover operator includes how to determine the location of the intersection and how to carry out the two aspects of gene exchange. The so-called crossover operation in genetic algorithm is to exchange some of the genes in two pairs of chromosomes in some way, so as to form two new individuals. The Cross method used in this article is partially matched crossover PMX. Its operation is to use randomly generated number to choose the corresponding two chromosome's cross string. Then we could use the cross string position to exchange operation exchange two corresponding, thus to achieve two kinds of conditions in the program. Assume a two father on the chromosome is divided into three regions, specific operation as follows:

$A = 1\ 2 \mid 3\ 4 \mid 5\ 6$

$B = 4\ 5 \mid 6\ 3 \mid 2\ 1$

First, select the exchange area of A and B (the exchange area is randomly selected) and the exchange is assumed to be the first exchange area.

$A_1 = 4\ 5 \mid 3\ 4 \mid 5\ 6$

$B_1 = 1\ 2 \mid 6\ 3 \mid 2\ 1$

Obviously, for A and B, the time is coincide so it is not a reasonable arrangement. The solution is to exchange the repeated arrangement that appear in the outside of the matching area in accordance with the position mapping relation in the matching region.

$A_2 = 4\ 5 \mid 3\ 1 \mid 2\ 6$

$B_2 = 1\ 2 \mid 6\ 3 \mid 5\ 4$

If the exchange area is selected in the second area, the exchange is different from the above. After the exchange method we could get the arrangement $A = 1\ 2 \mid 6\ 3 \mid 5\ 3$， $B = 3\ 5 \mid 3\ 4 \mid 2\ 1$ can still not be satisfied. Therefore, we should change the cross area of cross the same gene staining to:

$A_3 = 1\ 2 \mid 6\ 3 \mid 5\ 6$

$B_3 = 4\ 5 \mid 3\ 4 \mid 2\ 1$

We could detect the cross region of different genes on chromosome 6 in sixth, 4 genes in the chromosome is located in the first place. Thus, we can change the sixth order into genes 4 in the chromosome A, and change the first order into genes 6 in the chromosome B. It finally became a reasonable starting time order scheme.

$A_4 = 1\ 2\ |\ 6\ 3\ |\ 5\ 4$

$B_4 = 6\ 5\ |\ 3\ 4\ |\ 2\ 1$

(6) variation

The so-called mutation operation in genetic algorithm is to replace the gene locus on some loci in the individual chromosome code string, and to form a new individual. The crossover operation is the main method for the generation of new individuals. It determines the global search ability of genetic algorithm; auxiliary method and mutation operation is the generation of new individuals. But it is an essential method, because it determines the local search ability of genetic algorithm. The main purpose of variation is to maintain the population diversity, and to repair and supplement some genetic genes which may be lost in the process of selection and crossover. The commonly used mutation operators are: basic mutation operator, reversal operator and adaptive mutation operator, etc.. The mutation operator used for this question is the exchange of variation, that is, random selection of two genes on the chromosome of a simple swap.

(7) the setting of control parameters and the treatment of constraint conditions

The selection of control parameters in genetic algorithm is very important, and the different control parameters will have a great impact on the performance of genetic algorithm, which affects the convergence of the algorithm. In the genetic algorithm we must deal with constraints, but there is no general method to deal with various constraints. There are three methods that can be selected according to the specific problems, namely the search space is defined, the feasible solution transformation method and penalty function method.

2.4.2 Solving the Genetic Algorithm

The basic idea of genetic algorithm is that: from the initial population, the selection strategy is used to select individuals in the current population, and the crossover and mutation are used to produce the next generation. So the generation is keeping evolving, until the desired end is meet. A specific genetic algorithm steps and flow chart is shown below:

Step 1: Determine the encode method.

Step 2: Determine the fitness function. Fitness is a measure of the quality of the solution, which is the only basis for the genetic algorithm to perform various genetic operations in a population.

Step 3: Determine the genetic strategy. This will include the concern about the size of the population, the number of iterations and other auxiliary control parameters, selection, crossover, mutation and crossover probability, mutation probability and other genetic parameters.

Step 4: Randomly generate the initial population.

Step 5: Calculate the fitness value of the individual in the computing community.

Step 6: The selection, crossover and mutation operations are carried out in accordance with the genetic strategy to generate the next generation of groups.

Step 7: Determine whether the performance of the group meet the agreed termination conditions. If the end satisfies the condition then it ends. If not satisfied, then we should return to Step 6.

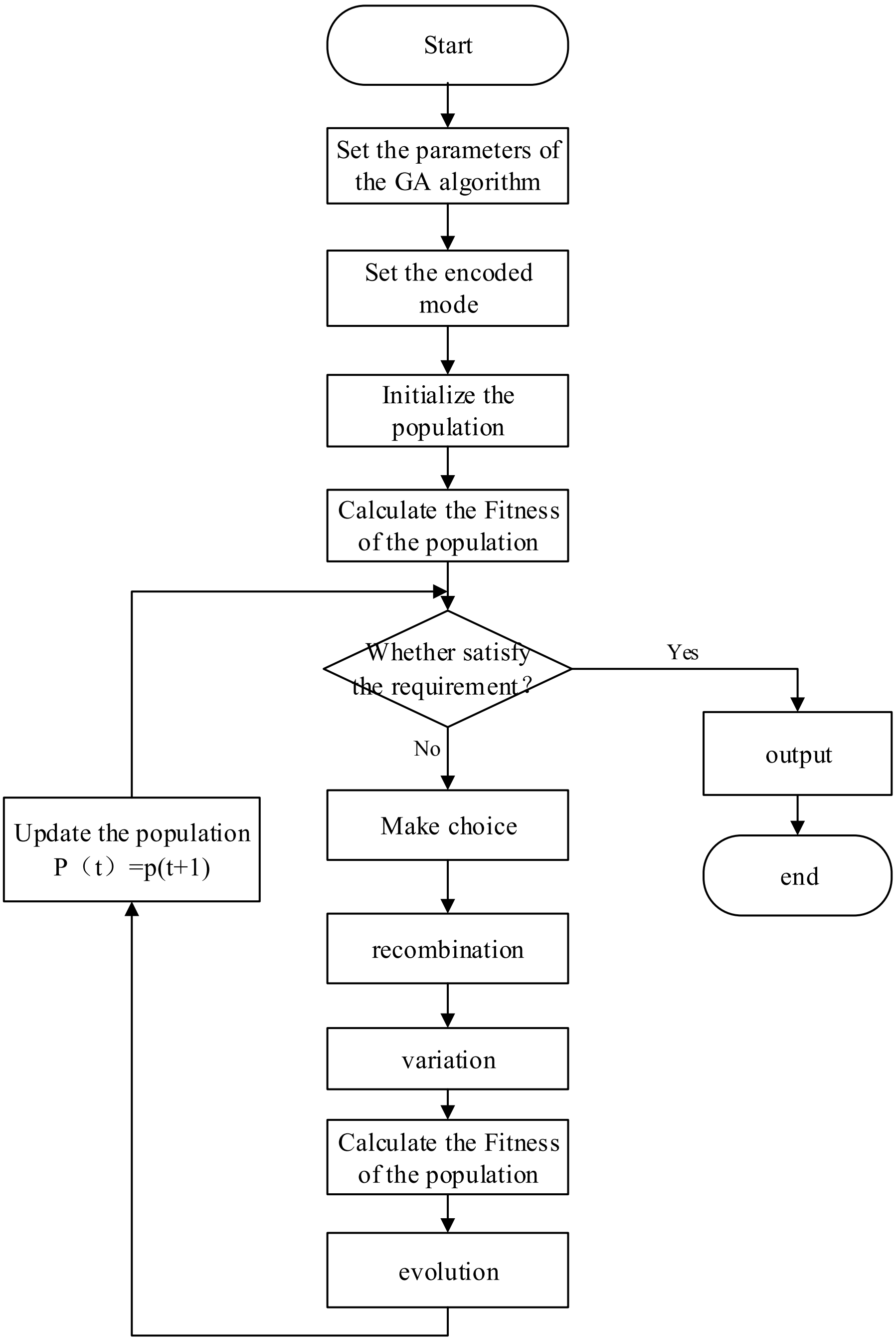


Figure 4 genetic algorithm flow chart

2.5 Application to the Triathlon race model

We use the result in model 1, and apply the Genetic Algorithm to find out the best start sequence of each group. The Genetic Algorithm above is realized by MATLAB, we choose the number of the population as 20, the number of the repeat is 100, the probability of cross is 0.8, and the probability of variation is 0.05. we set $\mu+\sigma$ to represent the time for each stage. The time table is as following :

Table 4 time table for every group in each sport

Groups	Swim/mins	T2	Bike/mins	T2	Run/mins
M OPEN(1)	27.61	10.63	96.51	4.52	70.925
F OPEN(2)	25.84	13.47	113.18	5.74	74.10
ATH(3)	26.97	15.28	128.34	6.39	91.37
M_PRE &F_PRO&M_PRO(4)	17.90	4.84	72.60	1.51	42.64
F PREMIER(5)	18.43	5.59	80.87	1.92	47.88
CLY(6)	29.46	11.24	101.15	5.59	85.08

From the application to the Genetic Algorithm, we could find first, the starting time of every group for each race as shown below:

Table 5 The Strat time of every group in each sport

Groups	Swim/mins	Bike/mins	Run/mins
M OPEN(1)	62.1674	100.4115	201.4399
F OPEN(2)	1.0000	40.3151	159.2351
ATH(3)	2.0000	44.2535	178.9825
M_PRE &F_PRO&M_PRO (4)	0	22.7470	96.8497
F PREMIER(5)	118.6720	142.6849	225.4781
CLY(6)	38.8275	79.5317	186.2757

For every group's each race's finishing time will be:

Table 6 the end time of every group in each sport

Groups	Swim	Bike	Run
M OPEN(1)	89.7817	196.9199	272.3550
F OPEN(2)	26.8451	153.4951	233.3351
ATH(3)	28.9739	172.5964	270.3550

M_PRE &F_PRO&M_PRO (4)	17.9032	95.3436	139.4923
F PREMIER(5)	137.0982	223.5598	273.3550
CLY(6)	68.2871	180.6830	271.3550

The race sequence for the 6 groups will be:

Pbest ={4 2 3 6 1 5}, which represents the start sequence of each group is as following: M_PRE &F_PRO&M_PRO->F_OPEN->ATH->CLY->M OPEN->F PREMIER.

Under this situation, the maximum time it takes to finish all the race will be: maxTime = 273.3550.

As shown below, the evolutionary process of genetic algorithm in Figures, the vertical axis represents each individual generation completion time. The horizontal axis represents the evolution algebra. From the graph, we can see that the minimum value of the objective function decreases with the evolution of the population, and finally converges to the extreme value.

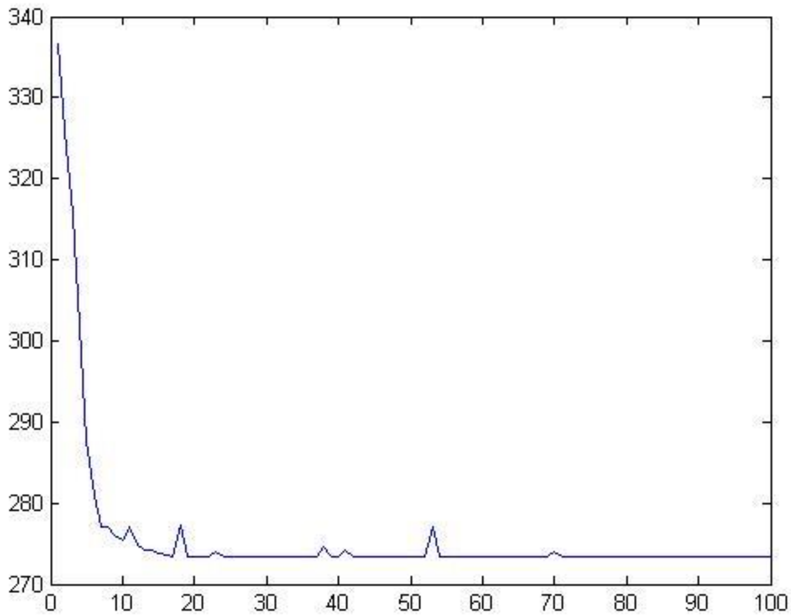


Figure 5 the evolutionary process of genetic algorithm 1

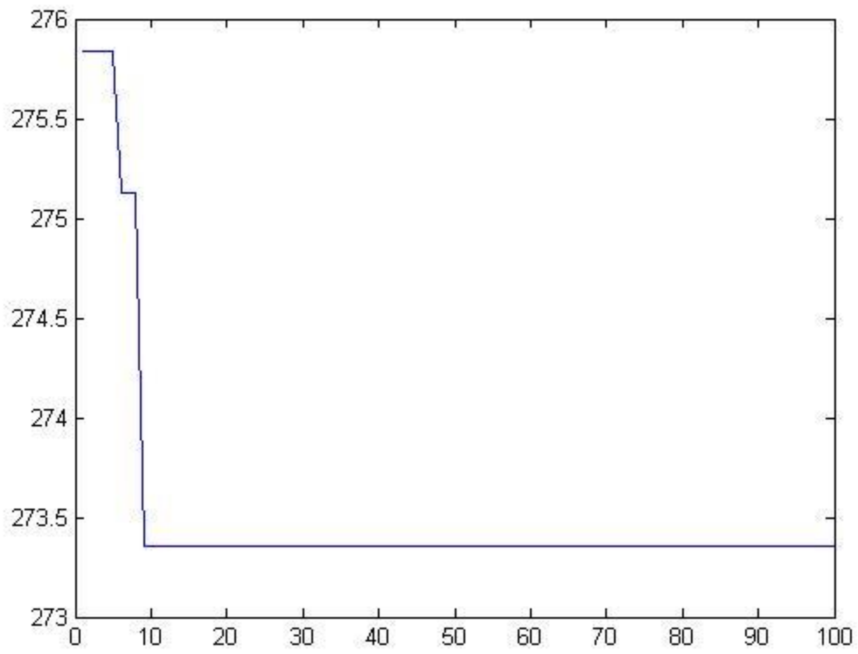


Figure 6 the evolutionary process of genetic algorithm2

Then, we can see clear time schedule using the Gantt Chart shown below:

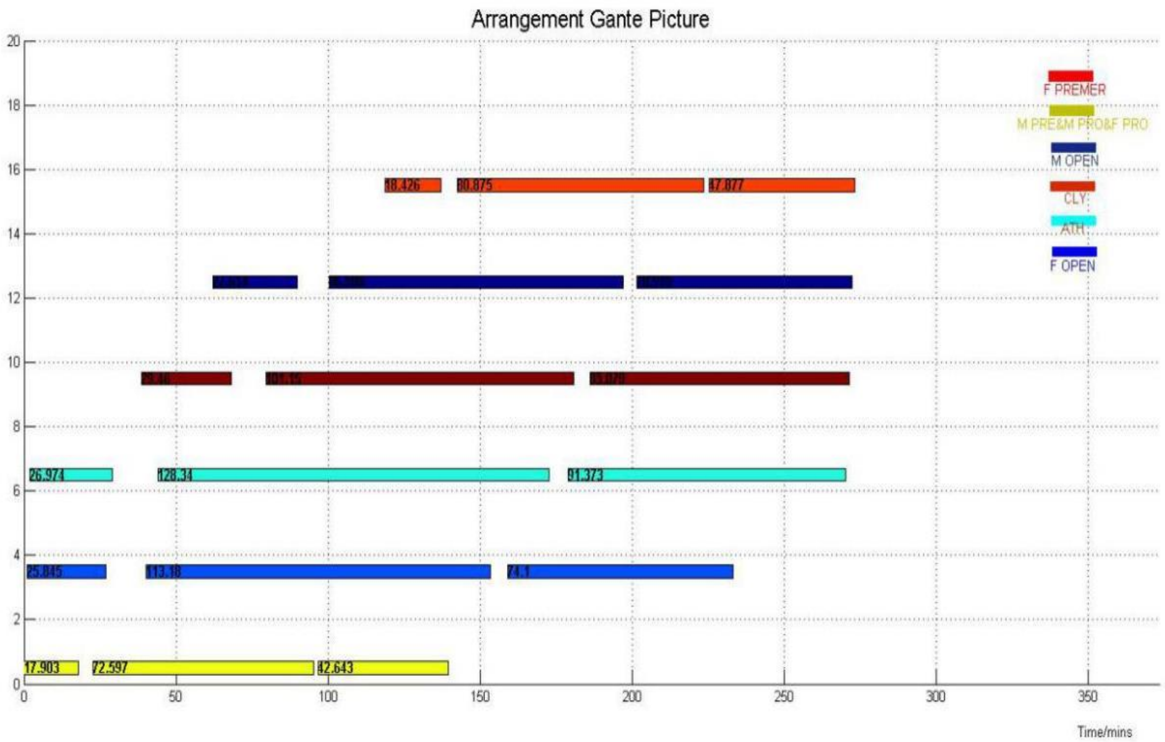


Figure 7 the arrangement Gantt picture

From the Figure above, we can see that each group of the people start at different time and they will not meet until they finish the competition, which guarantees that no mutual interference will happen.

When repeat more times doing this operation, we found out that the following orders are also fine with the question.

Pbest = {4 5 2 3 6 1}; max_time = 273.3550

Pbest ={2 3 6 1 5 4}; max_time = 273.3550

By repeating the genetic algorithm for a few times, we find the result is stable, and the max_time is very close to 27.3550 every time.

For each group of people, their numbers are quite different from each other. However, they have similar activity level. Thus we can let them start almost at the same time with only a few second intervals to make sure there is no congestion. Totally, we set 1 minute margin between each group for no congestion. The final event schedule is as following:

Table 7 the event schedule of the triathlon

Groups		Start time/mins	number
M_PRE &F_PRO&M_PRO	M_PRE &F_PRO&M_PRO[1]	0	20
	M_PRE &F_PRO&M_PRO[2]	0.01	19
F_OPEN	F OPEN[1]	1	20
	F OPEN[2]	1.01	20
	……	……	……
	F_OPEN[28]	1.28	18
ATH	ATH[1]	2	20
CLY	CLY[1]	38.83	20
	CLY[2]	38.84	17
M OPEN	M OPEN[1]	62.17	20
	M OPEN[2]	62.18	20
	……	……	20
	M_OPEN[67]	62.84	15
F PREMIER	F PREMIER[1]	118.67	11

VI. Model 3 : LP for the Optimization of Congestion and Road Closure Time

3.1 Basic Description of the Linear Programming

Linear programming is an effective tool to solve optimization problems, this tool is widely used in operational research and tends to be mature in theory. By applying existing conditions, such as labor force, funds and other resources for instance to the model, linear programming analysis relational database and achieves to achieve the maximum of outputs, profits. etc.

We apply this method to Part III.

1. Based on the information given by Model 2: a) optimized division of race; b) schedule of wave start time, we may get the fixed quantity of each participant's speed, in all groups and events, including long distance swimming, cycling and running.
2. Changing race distance of each event, based on the data from each participant's speed.
3. Drawing the conclusion to the minimum of total time that each group takes.

By adjusting race distance, we may get better result of traffic congestion and road closure time.

3.2 Analysis of the Needed Data

Given Condition. In the Model 1 and Model2, as traditional Olympic triathlon, the race distance is the following: a 1500-meter swimming, a 40K bike ride and a 10K run.

Analysis. By applying cluster analysis and GA model, we get each group's statistical time spent in each event.

Conclusion. With known quantity of time and distance, we calculate the average speed of each group in each event.

We set race event time as T (min), race event distance as S (K), and race event velocity as R (K/min). The velocity of race:

$$R = S / T \tag{3.1}$$

We take this formula into practice: the velocity of each group in each event shows elaborately in the table 8.

Table 8 Speed of Each Group

Velocity (K/min) / Group	Event		
	Swimming	**Biking**	**Running**
M OPEN	0.054319764	0.414471892	0.141013662
F OPEN	0.05804594	0.353420064	0.134949693
ATH	0.055609383	0.31166489	0.109442068
M_PRE &F_PRO&M_PRO	0.083783824	0.550989958	0.234507088
F PREMIER	0.081406132	0.494590861	0.208868791
CLY	0.050917146	0.395447175	0.117537417

Analogously, we set race event time as T' (mins), race event distance as S'(K), and race event velocity as R' (K/min) in our triathlon.

$$R' = S' / T' \tag{3.2}$$

Table 8 is applied to Model 3 base on two hypothesis:

1. All participants are in constant speed. Triathletes' stamina is neglected.
2. New triathlon' race distance is in a reasonable scope.

$$R' = R \tag{3.3}$$

The velocity of traditional Olympic triathlon is used in our Triathlon.

$$\begin{cases} R = S/T \\ R' = S'/T' \Rightarrow T' = S'{*}T/S \\ R = R' \end{cases} \tag{3.4}$$

To simplify the formula, we set a (min/K) as constant.

$$a = T/S \tag{3.5}$$

$$T' = S' \bullet a \tag{3.6}$$

By using the inferential reasoning formula and data analyzed above, we classify the T_{ij} as race event time of each group.(according to priority, i=1,2,3,4,5,6, which refers to each group and j=1,2,3, which refers to each race event).

Table 9 the race event time of each group in each sport

Race event time(T_{ij}) / Group(i)	Race Event(j)		
	Swimming(1)	Biking (2)	Running(3)
M OPEN(1)	T_{11}	T_{12}	T_{13}
F OPEN(2)	T_{21}	T_{22}	T_{23}
ATH(3)	T_{31}	T_{32}	T_{33}
M_PRE &F_PRO&M_PRO(4)	T_{41}	T_{42}	T_{43}
F PREMIER(5)	T_{51}	T_{52}	T_{53}
CLY(6)	T_{61}	T_{62}	T_{63}

By applying the data from Question, We set c – an additional fixed quantity, as the total transition time, which cannot be changed according to Questions.

$$c = T_1 + T_2 \tag{3.7}$$

3.3 The model establishment of linear programming

In the same way as Model 2, we set waiting time to measure if traffic congestion is improved and if road closure time is shorten, that is, the one with lowest ratio of waiting times to total time is the optimization model. Adopting group order of starting from Model 2, we have the hypothesis:

1. Each race event time should be more than or equal to the previous one;
2. The total time should be close to the previous total time;
3. Both contribute to the constricted condition of this arrangement.

The inequality of total race events for group M OPEN(1):

$$T_{11}+T_{12}+T_{13}\ge T_{1total} \tag{3.8}$$

The inequality of total race events for group F OPEN(2):

$$T_{21}+T_{22}+T_{23}\ge T_{2total} \tag{3.9}$$

The inequality of total race events for group ATH(3):

$$T_{31}+T_{32}+T_{33}\ge T_{3total} \tag{3.10}$$

The inequality of total race events for group M_PRE &F_PRO&M_PRO(4):

$$T_{41}+T_{42}+T_{43}\ge T_{4total} \tag{3.11}$$

The inequality of total race events for group M_PRE &F_PRO&M_PRO(4):

$$T_{51}+T_{52}+T_{53}\ge T_{5total} \tag{3.12}$$

The inequality of total race events for group M_PRE &F_PRO&M_PRO(6):

$$T_{61}+T_{62}+T_{63}\ge T_{6total} \tag{3.13}$$

Besides, the quantity of race event time cannot be negative:

$$T_{ij}\ge 0\left(i=1,2,3,4,5,6;\ j=1,2,3\right) \tag{3.14}$$

Moreover, based on the physical limit of human beings, every race event distance cannot be bottomless. According to the documented material from website, the extreme race distance of triathlon include: swimming 3.8K, biking 120K, and running 42.195K. Therefore, we sets the maximum of possible event race distance.

$$\begin{cases} S_1\le 3.8 \\ S_2\le 120 \\ S_3\le 42.195 \end{cases} \tag{3.15}$$

In this way, we set the total time of each group in our triathlon as Tti.', which is the combination of each race event time and total transition time of each group.

$$T_{ti}' = \mathrm{T_i}'+\mathrm{c} \tag{3.16}$$

According to the optimized arrangement of starting order (M_PRE &F_PRO&M_PRO -> F OPEN-> ATH-> CLY-> M OPEN-> F PREMIER）, we classify the race distance of each event as variable. Then by employing the algorithm of model 2, we calculate the total time of our triathlon. Therefore, we get total time's linear equation on each race event distance.

Calculate the result of optimized total time in our triathlon based on the following constricted conditions:

- Obey model 2 optimized arrangement of starting order;
- The distance of each event in our new triathlon is not less than that in traditional Olympic triathlon; furthermore, it is not more than that in the extreme race.

Thence, we get the result of optimized total time in our triathlon, and compare it with the result shown in model 2, which serves as an effective vehicle to confirm whether adjusting race event distance can optimize the road closure and total time.

According to the specific analysis and prerequisites mentioned above, we list these following inequalities and object function, Each inequality should be satisfied to meet the standard of profession triathlon and the Limits in reality.

Object function:

$$\min f = 17.98 \bullet S1 + 3.21 \bullet S2 + 9.14 \bullet S3 + 26.67 \tag{3.17}$$

Constrained condition:

$$\begin{cases} 18.41 \bullet S_1 + 2.41 \bullet S_2 + 7.09 \bullet S_3 + 15.14 \geq 210.19 \\ 17.23 \bullet S_1 + 2.83 \bullet S_2 + 7.41 \bullet S_3 + 19.21 \geq 232.33 \\ 17.98 \bullet S_1 + 3.21 \bullet S_2 + 9.14 \bullet S_3 + 21.67 \geq 268.36 \\ 11.94 \bullet S_1 + 1.81 \bullet S_2 + 4.26 \bullet S_3 + 6.35 \geq 139.49 \\ 12.28 \bullet S_1 + 2.02 \bullet S_2 + 4.79 \bullet S_3 + 7.51 \geq 154.68 \\ 19.64 \bullet S_1 + 2.53 \bullet S_2 + 8.51 \bullet S_3 + 16.83 \geq 232.53 \\ S_1, S_2, S_3 \geq 0 \\ S_1 \leq 3.8 \\ S_2 \leq 120 \\ S_3 \leq 42.195 \end{cases} \tag{3.18}$$

3.4 The result of the Model

With the help of MATLAB, our linear programming models get the following results.

Optimized race distance shows as follows:

Table10 the race event time of each group in each sport

Swimming(km)	Biking(km)	Running(km)
3.7290	44.5888	4.0024

Under this situation, the total time of our triathlon is 273.3600 min

the time consumption comparison shows as follows:

Table11 the race event time before and after adjustment

Race event time(T_{ij}) / Group(i)	Time	
	Before	After
M OPEN(1)	210.19	219.76
F OPEN(2)	232.33	239.28
ATH(3)	268.36	268.36

M_PRE &F_PRO&M_PRO(4)	139.49	148.85
F PREMIER(5)	154.68	162.63
CLY(6)	232.53	236.88

Comparing the data in the table above, each group spend more time on each sport correspondingly. Through optimization, the total time of the whole triathlon remains appropriately constant.
Less traffic congestion means less time in waiting and more efficient in racing. Avoiding road closure and congestion, this model keep the total racing time almost the same while prolongs each group's racing time. That is, we use the minimum waiting time to comfort road capacities.

VII. Conclusions

To draw a conclusion, our paper firstly extract the classification parameter aims to the previous Triathlon data, uses cluster analysis to divide athletes into groups. The division, for one hand, will allow the athletes to compete with the athletes at similar levels and diminish the interfere between them. It can therefore reduce the total race time. For another, the division could reduce the number of athletes in one group and it could solve the problem of road congestion. Secondly, our paper uses the Genetic Algorithm to solve the problem of the starting sequence of all the groups. This will allow us to get the shortest time for the whole race under the condition without interference between athletes and road congestion. The final determination for the starting order will be as following:

Table 12 the final event schedule

Groups		Start time/mins	number
M_PRE &F_PRO&M_PRO	M_PRE &F_PRO&M_PRO[1]	0	20
	M_PRE &F_PRO&M_PRO[2]	0.01	19
F_OPEN	F OPEN[1]	1	20
	F OPEN[2]	1.01	20
	……	……	……
	F_OPEN[28]	1.28	18
ATH	ATH[1]	2	20
CLY	CLY[1]	38.83	20
	CLY[2]	38.84	17
M OPEN	M OPEN[1]	62.17	20
	M OPEN[2]	62.18	20
	……	……	……
	M_OPEN[67]	62.84	15
F PREMIER	F PREMIER[1]	118.67	11

Finally, we analysis the relationship between each race's distance and the total races time in the paper. On the basis of model 2's optimal starting order, we could get the optimal distance for each race by applying the linear programming model. This will allow us to achieve the highest utilization rate of the total races time under the condition without interference between athletes and road congestion. After the optimization, the distance for each race will be: swimming: 3.7290km, biking:44.5888km, running:4.0024km. The total race time after the optimization will be 273.36mins.

VIII.Strengths and weaknesses

Strengths:

- ✓ Derive mean and sigma from raw data in model 1 to show the features of data, we consider more comprehensively than only deriving mean, with better representation of overall data.
- ✓ We diminish the categories of classification effectively through Cluster Analysis, which serves for organizing the competition, reducing the road congestion, and reduce the time needed of road closure.
- ✓ We apply Genetic Algorithm to optimize the order of competition, for the convergence of algorithm, we derive the optimum solution in the shortest amount of time.
- ✓ The generality is strong in model2, it can be applied not only to this problem, but also to daily pipeline problems.
- ✓ We use the variable-controlling method in model 3 to analyze the influence on time and congestion exerted by the distances of swimming, biking, running, making the result more scientific and reliable.
- ✓ The overall model accords with the real situation, with strong generalization; can be applied to general scheduling problem.

Weaknesses

- ✓ The model adopts the interval between the departure of every groups to be several seconds, generally leave 1 minute for the interval of departure of a whole category. We do the assumption based on experience, therefore lack of the verification of mathematic.
- ✓ For the optimization of distances, because the question does not give corresponding constrict condition, we have to make assumptions, which are possible to have error compared to real situation.

References:

[1] https://en.wikipedia.org/wiki/Triathlon

[2] Holsapple C.W. Jacop V.S., Pacart R., Zaveri J.S. A genetics-based hybrid scheduler for gen er ating static schedules in flexible manufacturing contexts [J]. IEEE Transactions on Systems, man a nd Cyberneties, 1993, 23(4):953-971.

[3] Sim S,K.Yeo K.T. Lee W.H.. An expert neural network system for dynamic job shop schedu li ng [J]. International Journal of Production Researeh, 1994, 32(8):1795-1773.

[4] Lee C.Y. Piramuthu S. Tsai Y.K. Job shop scheduling with a genetic algorithm and machine le arning[J]. International Journal of Production Researeh, 1997, 35(4):1171-1191.

[5] Jian A.K. Elmaraghy H.A. Production scheduling rescheduling in flexible manufacturing [J]. In teractional Journal of Production Research, 1997,35(l):281-309.

[6] https://en.wikipedia.org/wiki/Genetic_algorithm

[7] https://en.wikipedia.org/wiki/Cluster_analysis

Appendix

Part of code for the genetic algorithm

```
function [Pbest,maxtime,Lc1,Lc2]=JSPGA2(M,N,Pc,Pm,T,P)
Pbest=zeros(1,size(P,2));
Lc1=zeros(1,N);
Lc2=zeros(1,N);
farm=cell(1,M);
if mod(floor(M*Pc),2)==1
   error('Exchange Rate illegal!');
end
for i=1:M
   farm{i}=randperm(6);
end
counter=0;

while(counter<N)
   newfarm=cell(1,floor(M*Pc));
   Fmp=randperm(M);
```

```
nochangenum=randperm(ceil(M*(1-Pc)));
unchange=Fmp(nochangenum);
Fmp(nochangenum)=[];
for i=1:size(Fmp,2)/2
   Fa=farm{Fmp(2*i-1)};
   Ma=farm{Fmp(2*i)};
   pp=randperm(3);
   a=pp(1);
   Za=Fa;
   Zb=Ma;
   Za(2*a-1:2*a)=Ma(2*a-1:2*a);
   Zb(2*a-1:2*a)=Fa(2*a-1:2*a);
   [Za,Zb]=legalson(Fa,Ma,Za,Zb,a);
   newfarm{2*i-1}=Za;
   newfarm{2*i}=Zb;
end
tempFarm=[farm,newfarm,farm{unchange}];
FITNESS=zeros(1,2*M);
fitness=zeros(1,M);
for i=1:2*M
   X=tempFarm{i};
   FITNESS(i)=cost2(T,X);
end
Ser=randperm(2*M);
for i=1:M
   f1=FITNESS(Ser(2*i-1));
   f2=FITNESS(Ser(2*i));
   if f1<=f2
      farm{i}=tempFarm{Ser(2*i-1)};
      fitness(i)=FITNESS(Ser(2*i-1));
   else
      farm{i}=tempFarm{Ser(2*i)};
      fitness(i)=FITNESS(Ser(2*i));
```

```
        end
    end
    for i=1:M
        if Pm>rand;
            Y=farm{i};
            Ix=randperm(6);
            temp=Y(Ix(1));
            Y(Ix(1))=Y(Ix(2));
            Y(Ix(2))=temp;
            farm{i}=Y;
        end
    end
    minfitness=min(fitness);
    meanfitness=mean(fitness);
    Lc1(counter+1)=minfitness;
    Lc2(counter+1)=meanfitness;
    pos=find(fitness==minfitness);
    Pbest=farm{pos(1)};
    counter=counter+1;
end
figure(1);
[maxtime,Ts,Te]=cost2(T,Pbest);
DrawGante(Ts,Te,T,Pbest);
figure(2);
plot(1:N,Lc1);
figure(3);
plot(1:N,Lc2);
function [maxtime,Ts,Te]=cost2(T,P)
[m,n]=size(T);
Ts=zeros(m,n);
Te=zeros(m,n);
Ts(P(1),1)=0;
for i=1:m
```

```
  for j=1:2:5
     if i>1 && j==1
          WAIT_TIME=max([(T(P(i-1),j)+T(P(i-1),j+1))-(T(P(i),j)+ T(P(i),j+1)),...
                     (T(P(i-1),j)+T(P(i-1),j+1))-(T(P(i),j)+  T(P(i),j+1))  +  (T(P(i-1),7)+T(P(i-1),8))-
(T(P(i),7)+T(P(i),8)),...
                     (T(P(i-1),j)+T(P(i-1),j+1))-(T(P(i),j)+  T(P(i),j+1))  +  (T(P(i-1),7)+T(P(i-1),8))-
(T(P(i),7)+T(P(i),8))+(T(P(i-1),j+2)+T(P(i-1),j+3))-(T(P(i),j+2)+T(P(i),j+3)),...
                     (T(P(i-1),j)+T(P(i-1),j+1))-(T(P(i),j)+  T(P(i),j+1))  +  (T(P(i-1),7)+T(P(i-1),8))-
(T(P(i),7)+T(P(i),8))+(T(P(i-1),j+2)+T(P(i-1),j+3))-(T(P(i),j+2)+T(P(i),j+3))+(T(P(i-1),9)+T(P(i-
1),10))-(T(P(i),9)+T(P(i),10)),...
                     (T(P(i-1),j)+T(P(i-1),j+1))-(T(P(i),j)+  T(P(i),j+1))  +  (T(P(i-1),7)+T(P(i-1),8))-
(T(P(i),7)+T(P(i),8))+(T(P(i-1),j+2)+T(P(i-1),j+3))-(T(P(i),j+2)+T(P(i),j+3))+(T(P(i-1),9)+T(P(i-
1),10))-(T(P(i),9)+T(P(i),10))+(T(P(i-1),j+4)+T(P(i-1),j+5))-(T(P(i),j+4)+T(P(i),j+5))...
            ]);
           if WAIT_TIME<=0
            Ts(P(i),(j+1)/2)=Ts(P(i-1),(j+1)/2)+1;
           else
            Ts(P(i),(j+1)/2)=Ts(P(i-1),(j+1)/2)+WAIT_TIME+1;
           end


     end
     if i==1 && j>1
        Ts(P(i),(j+1)/2)=Ts(P(i),(j+1)/2-1)+T(P(i),j-2)+T(P(i),j-1)+T(P(i),j+4)+T(P(i),j+5);
     end
     if i>1 && j>1
         Ts(P(i),(j+1)/2)=Te(P(i),(j+1)/2-1)+T(P(i),j+4)+T(P(i),j+5);


     end
     Te(P(i),(j+1)/2)=Ts(P(i),(j+1)/2)+T(P(i),j)+T(P(i),j+1);
  end
end
maxtime=Te(P(m),3);
```