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

Team Control Number

37132

Problem Chosen

A

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

2015
Mathematical Contest in Modeling (MCM/ICM) Summary Sheet

Summary

Our goal is how to conduct medical demand forecast based on the serious Ebola epidemics and propose the distribution strategy of medicine and vaccine from the view of optimization and stimulation.

First of all, we consider a SIR model and formulate an optimal control problem with vaccination and treatment as controls. Our aim is to find the optimal combination of vaccination and treatment strategies that will minimize the cost of the two control measures as well as the number of infectious people.

Based on SIR model combining with optimal levels of the two controls, the collaborative relationship between the epidemic diffusion network and the emergency medical delivery is established by medicine and vaccine demand forecast. Besides, it can be seen that once the relative cost of each of the control measures is stimulated as real situation, how the epidemic diffusion and medical demand is influenced by the optimal combination of vaccination and treatment strategy and therefore influence the distribution scheme.

Then, according to medical demand and urgency degree of the affected area, an optimal medicine and vaccine distribution model is proposed with the background of the epidemic outbreak in one certain region.

In this section, we analyzes the characteristics of emergency logistics, designs the system structure of emergency medical and vaccine distribution, and establishes an optimization model for vehicle scheduling in emergency logistics. It adopts genetic algorithm which has better route selection and shorter time allocation. We designs the nonlinear fitness function, and employs nonlinear ranking selection, blend crossover and blend mutation to solve the model. A numerical example is presented to show the effectiveness and feasibility of this algorithm.

From our model, this problem is divided into two parts: one is medical demand forecast for every infectious area, another is the multiple distribution. It is analyzed among different epidemic areas reasonably according to the time-varying demand of each epidemic area for the purpose of optimizing the emergency medicine scheduling.

Keywords: vaccination and treatment; optimal control; vehicle schedule; emergency medical distribution; Genetic Algorithm

February 9, 2015

Summary

Our goal is how to conduct medical demand forecast based on the serious Ebola epidemics and propose the distribution strategy of medicine and vaccine from the view of optimization and stimulation.

First of all, we consider a SIR model and formulate an optimal control problem with vaccination and treatment as controls. Our aim is to find the optimal combination of vaccination and treatment strategies that will minimize the cost of the two control measures as well as the number of infectious people.

Based on SIR model combining with optimal levels of the two controls, the collaborative relationship between the epidemic diffusion network and the emergency medical delivery is established by medicine and vaccine demand forecast. Besides, it can be seen that once the relative cost of each of the control measures is stimulated as real situation, how the epidemic diffusion and medical demand is influenced by the optimal combination of vaccination and treatment strategy and therefore influence the distribution scheme.

Then, according to medical demand and urgency degree of the affected area, an optimal medicine and vaccine distribution model is proposed with the background of the epidemic outbreak in one certain region.

In this section, we analyzes the characteristics of emergency logistics, designs the system structure of emergency medical and vaccine distribution, and establishes an optimization model for vehicle scheduling in emergency logistics. It adopts genetic algorithm which has better route selection and shorter time allocation. We designs the nonlinear fitness function, and employs nonlinear ranking selection, blend crossover and blend mutation to solve the model. A numerical example is presented to show the effectiveness and feasibility of this algorithm.

From our model, this problem is divided into two parts: one is medical demand forecast for every infectious area, another is the multiple distribution. It is analyzed among different epidemic areas reasonably according to the time-varying demand of each epidemic area for the purpose of optimizing the emergency medicine scheduling.

Keywords: vaccination and treatment; optimal control; vehicle schedule; emergency medical distribution; Genetic Algorithm

1 Introduction

The Ebola outbreak in West Africa was first reported in March 2014, and has rapidly become the deadliest occurrence of the disease since its discovery in 1976. It spreads between humans by direct contact with infected blood, bodily fluids or organs, or indirectly through contact with contaminated environments. Even funerals of Ebola victims can be a risk, if mourners have direct contact with the body of the deceased. To defeat Ebola, many countries have put a lot of energy to the study of treatment drug and vaccine.

1.1 Restatement of the Problem

Therefore, how to swiftly deal with the serious epidemics, reasonably establish emergency distribution system and effectively strengthen the supply of medicine and vaccine are becoming an arduous and complicated project. We analyze this problem from the view of optimization and stimulation.

- Optimization

  Based on classic epidemic diffusion model, we studied with the objective of scientifically optimizing the emergency medical distribution system where the spread of Ebola, the quantity of the medicine demanded and possible feasible delivery scheme are included.

- Stimulation

  The mixed integer programming is employed to research multiple emergency distribution networks among distribution center and demand point and then solved by the genetic algorithm.

1.2 Literature Review

Emergency logistics research has only begun to attract attention in recent years.

Anwen Lu [1] with other researchers discussed the essentiality of emergency goods distribution under the urgent circumstances, established road transport logistics and distribution model under emergency situations.

Jin Li [2] proposed the multi-resource and multi-disaster-place scheduling model based on network optimization in graph theory and linear programming optimization.

Jun Tian [3] made use of the triangular fuzzy numbers to describe the emergency demands for supplies as well as thinking of the prioritized condition.

Sen Chen [4] built a combinatorial optimization model combining the road network and vehicle routing.

1.3 Our Model Overview

In Section 1, in order to forecast medical demand precisely, we address the question of how to optimally combine the vaccination and the treatment strategies such that the cost of the two interventions is minimized while the disease is eradicated within specified period as soon as quickly.

The section 1 is organized as follows.

(i) Present the SIR model to be investigated

(ii) Formulate an optimal control problem subject to the SIR model, characterize the optimal controls, and derive the optimality system

(iii) Conduct our demand forecast based on the resulting optimality system numerically

Then, in section 2, according to the epidemic diffusion rule and the different urgency degree of the affected areas, an optimal medicine and vaccine distribution approach is studied with the background of the epidemic outbreak in one certain region. Furthermore, a dynamic vaccine distribution model is proposed based on the epidemic diffusion rule and clustering approach.

Numerical study indicates that the proposed approach can make an outstanding contribution to controlling the affected areas with relative high degree of urgency, also show the effectiveness and feasibility of genetic algorithm.

2 Assumptions and Justifications

The emergency medicine and vaccine distribution is oriented by the demand of infectious sites, which aims at meeting the need of the infectious sites and shortens the distribution time as soon as possible in order to cut down the death rate. We make the following definitions and descriptions:

- **Vaccine is anti-virus effectively.** Those who inject vaccine will not become infectious in short time.
- **Supply depot provides several types of vehicles.** All the vehicles are sufficient for emergency supplies distribution and each vehicle has a serial number to mark it.
- **Vehicle need not return to starting point (depot)immediately.** After finishing a distribution assignment, vehicle heads for the infectious site for next order.
- **Points at the same level have nothing to do with one another in distribution.** There are neither medicine (treatment drugs) flows nor vaccine flows from one demand site supplying to another. All items are provided by supply point.
- **Loading efficiency at supply points is equal to the unloading efficiency at demand points.**

3 Section I: Medical Demand Forecast Model

In this section, we studied optimal combination of vaccination and treatment strategies for driving infectious diseases with cure and vaccine towards eradication within a specified period. We considered an SIR model with varying size population using vaccination and treatment as control measures.

Based on SIR model combining with optimal levels of the two controls, the collaborative relationship between the epidemic diffusion network and the emergency medical delivery is established by researching the medicine and vaccine demand forecast.

3.1 SIR Model formulation

Suppose S represents the proportion of the members of the population susceptible to the disease, I represents the proportion of the number of individuals who are infected, and R represents the proportion of the number of individuals who are removed due to vaccination or recovery from the disease which confers permanent immunity to reinfection.

We now consider the SIR model below:

$$\begin{aligned} \dot{S} &= b - \beta SI - dS - u_1 S \\ \dot{I} &= \beta SI - u_2 I - dI - \alpha I \\ \dot{R} &= u_1 S + u_2 I - dR \end{aligned} \tag{1}$$

In the model, parameter descriptions list as follows:

- α : disease-induced death rate
- β : average number of people with which every infectious person effectively contact
- u_1 : the proportion of the susceptible people that is vaccinated per unit time
- u_2 : the proportion of the infectious people that are treated per unit time
- b : the recruitment rate
- d : the natural death rate

SIR model can be vividly shown in Fig.1:

The total population N(t) can be obtained from N(t)=S(t)+I(t)+R(t).

$$\dot{N} = b - Nd - \alpha I \tag{2}$$

Here, it is important to note that in the absence of the disease, $N(t) \to \frac{b}{a}$

Moreover, under the dynamics described by Eqs.(1)and(2), the region

$\Omega = \left\{ x = (S, I, N) \epsilon R_+^3 | S \geqslant 0, I \geq 0, S + I \leqslant N \leqslant \frac{b}{d} \right\}$

is positively invariant. Therefore, for initial starting point ϵR_+^3, the trajectory lies in Ω. Thus, we can restrict our analysis to the region Ω.

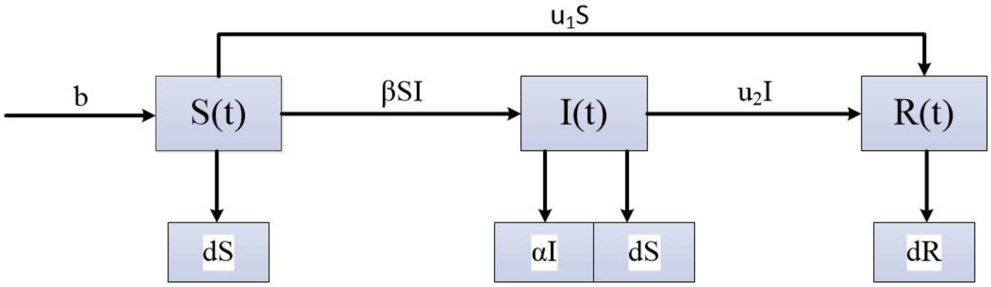

Figure 1: SIR flow chart

3.2 Dynamic Demand Forecast under optimal control

3.2.1 Optimization description

We define our objective functional as:

$$Z = \min_{u1,u2} \left\{ I(T) + \frac{1}{2} \int_0^T (C_1 u_1^2 + C_2 u_2^2) dt \right\} \tag{3}$$

subject to system above with appropriate states initial conditions while the control set U is measurable and it is defined as

$$U = \left\{ \left(u_1^t, u_2^t \right) | 0 \leq u_1 < u_{1max \leq 1}, 0 \leq u_1 < u_{1max \leq 1}, t \epsilon \left[0, T \right] \right\}$$

where C_1 and C_2 are the relative weights attached to the cost of vaccination and treatment respectively, u_{1max} is maximum attainable value for u_1 and u_{2max} is maximum attainable value for u_2.

The control u_1 is the proportion of the susceptible that is vaccinated per unit time while the control u_2 is the proportion of the infectious people that is treated per unit time. Thus, v and u_2 lie between 0 and 1 while u_{1max} and u_{2max} will depend on the amount of resources available to implement each of the control measures.

The weights C_1 and C_2 will depend on the relative importance of each of the control measures in mitigating the spread of the disease as well as the cost (human effort, material resources etc.) of implementing each of the control measures per unit time. Thus, the terms $C_1 u_1^2$ and $C_2 u_2^2$ describe the costs associated with vaccination and treatment respectively. The vaccination cost could include the cost of the vaccine, the vaccine storage cost, other related overheads, etc. The treatment cost could include the cost of the medical tests and diagnosis, drug cost, hospitalization cost, etc.

The term $I(T)$ is so constructed as a terminal cost because it seems more realistic to minimize the number of infective at end of the implementation of the control programme than at each time unit within the implementation period.

3.2.2 Function analysis

Our goal is to characterize an optimal control (u_1^*, u_2^*) which minimizes the cost of the vaccination and the cost of the treatment over the specified time interval as well as minimizes the number of infectious people at terminal time.

According to the authoritative results,our resulting optimality system is:

Substituting $u_1 = u_1^* \ and \ u_2 = u_2^*$ and solving for the optimal pair, we obtain

$$u_1^* = \frac{\lambda_1 S}{C_1} \quad , \qquad u_2^* = \frac{\lambda_2 I}{C_2}$$

Now, our resulting optimality system is:

$$\begin{cases} \dot{S} = b - \beta SI - dS - u_1^* S, \\ \dot{I} = \beta SI - u_2^* I - dI - \alpha I, \\ S(0) = S_0, \\ I(0) = I_0, \\ \dot{\lambda}_1 = \lambda_1(\beta I + d + u_1^*) - \lambda_2 \beta I, \\ \dot{\lambda}_2 = \lambda_1 \beta S - \lambda_2(\beta S - u_2^* - d - \alpha), \end{cases} \tag{4}$$

3.3 Simulation results and discussion

Numerical solutions to the optimality system are executed using MATLAB with the following realistic hypothetical parameter values and initial conditions:

Table 1: SIR Model Parameter

b	d	α	β	S(0)	I(0)	N(0)
0.03	0.2	0.1	0.75	0.95	0.05	1.0

We use this prediction model to analyze the supply prediction of medicine and vaccine. Based on real Ebola epidemic distribution, we use MATLAB to randomly generate data which belongs to normal distribution. We define children, pregnant women and the older as low immunity group.The initial epidemic distribution at early prevention is in Table 2.

After collecting and analyzing large quantities of prevalence data about Ebola and other infectious disease data, we finally determine that the $C_1 : C_2 = 1 : 4$ is closer to the real situation.

We used the forward-backward sweep scheme; starting with an initial guess for the optimal controls u_1 and u_2. The state system is solved forward in time while the solution to the states together with the initial guess for the controls are then used to solve the co-state system backward in time. Subsequently, we determine controls u_1 and u_2 as given above while the iteration continues until convergence is achieved. The results from our simulations are displayed in the Fig.2, Fig.3 and 4.

Numerical simulations of the resulting optimality system showed that, if it is more expensive to treat than to vaccinate, more resource must be put into vaccination.This case resulted in a rapid reduction in the susceptible people as well as an appreciable reduction in the number of infective people. Nevertheless, the case showed that the optimal way to drive the epidemic towards eradication within the specified period is to use more of the vaccination control and less of the treatment control initially to drive the epidemic to below certain threshold after which we can then apply less of vaccination control and more of the treatment control.

Based on these curves, we could overview the trend for S,I,R parameters in SIR model and make forecast on the demand for every site at next time. The results are shown in Table 3 and Table 4.

Remark 1:

1-th represents if we have carried out treatment currently, what quantity the demand next period will be. Similar to 1-th, 2-th represents what quantity the demand after two cycles will be.

Table 2: Raw epidemic data

Area	Total population	Susceptible	Infected persons	Low immunity
area-1	1442	1327	115	326
area-2	867	834	33	194
area-3	1686	1550	136	442
area-4	1796	1615	181	219
area-5	1107	1041	65	353
area-6	2119	1976	143	322
area-7	2138	1976	163	208
area-8	1505	1485	20	83
area-9	1659	1631	28	294
area-10	1698	1570	129	199
area-11	1505	1425	80	361
area-12	1925	1790	134	351
area-13	1406	1265	141	469
area-14	2509	2373	136	359
area-15	1610	1445	165	236
area-16	1679	1546	133	338
area-17	2086	1927	160	199
area-18	1564	1524	40	298
area-19	1561	1462	99	295
area-20	1259	1167	92	140

Table 3: Forecasting the demand for vaccine

Area	1	2	3	4	5	6	7	8	9	10
1-th	109	112	99	81	40	77	70	58	177	98
2-th	196	165	62	48	50	34	31	53	126	61
Area	11	12	13	14	15	i6	17	18	i9	20
1-th	210	98	98	65	91	99	76	166	118	252
2-th	327	53	81	21	62	88	35	200	83	364

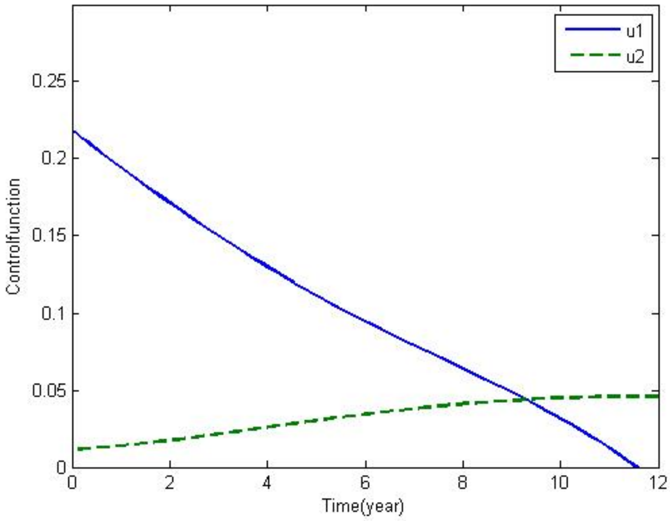

Figure 2: The control function u_1 and u_2 for $C_1 : C_2 = 1 : 4$

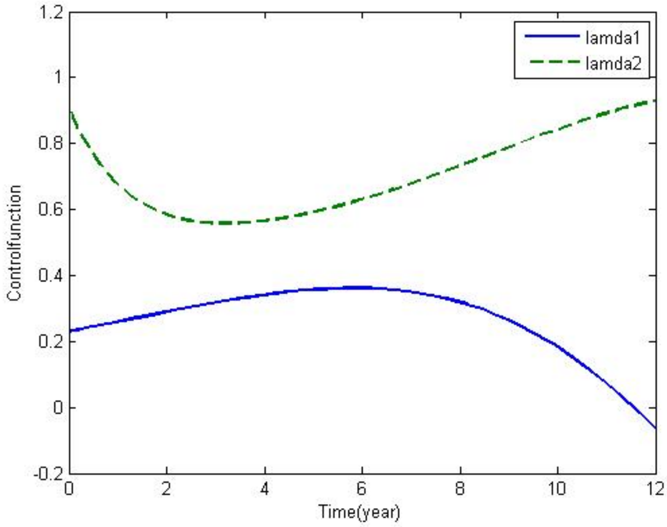

Figure 3: The control function λ_1 and λ_2 for $C_1 : C_2 = 1 : 4$

Table 4: Forecasting the demand for medicine

Area	1	2	3	4	5	6	7	8	9	10
1-th	138	40	163	217	78	172	196	25	34	155
2-th	69	20	82	109	39	86	98	12	17	78
Area	11	12	13	14	15	16	17	18	19	20
1-th	96	161	145	163	198	160	176	48	119	110
2-th	48	80	85	82	100	80	96	24	60	55

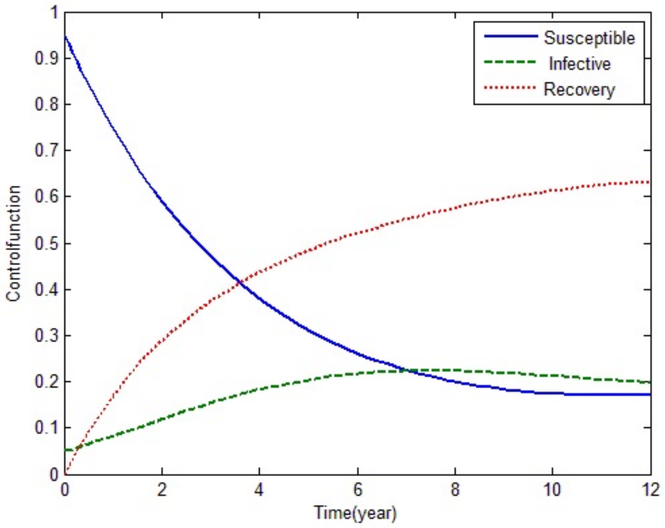

Figure 4: The population of S and I,R for $C_1 : C_2 = 1 : 4$

4 Section II: Emergency medical distribution Model

4.1 Problem description

Due to *Single Parent Genetic algorithm* replaces traditional crossover operation with a genetic recombinant algorithm on a single chromosome, simplify hereditary manipulation and improve calculating efficiency. The problem to converge towards local optima will be solved.

In this section, we use *Single Parent Genetic algorithm* to solve the objective function. Because in actual logistics distribution network, every logistic center usually have respond to its nearly demanding point, this algorithm first use *expand area cover method* to divide N demanding points into M groups(corresponding to M distribution centers). Every elements in set i is a demanding point which the i_{th} distribution center has responsibility to it. Elements can recur in different sets. And if an element recur in multiple sets, it is said that this point can be supplied by any one of the center.

Fig.5 shows that, there is 3 Logistics center p_0,p_1,p_2, 20 demanding points(0∼19), we adopt *expand area cover method* to group each point. In this figure, demanding point 6 locates in the intersection of three set, so it can be delivered by any of center. Similarly, demanding 5 locates in the intersection of p_0 and p_2, so either p_0 or p_2 should deliver medicine to it. However, point 0 just can be supplied by p_0.

As Fig.6 shown, There is several demanding points scattered in the West Africa. We regard Africa medicine delivery problem as a multi-center distribution problem.

Using *expand area cover method* can convert multi-center distribution problem into single-center distribution problem. We use Set Divide to match each demanding point with distribution center, and then for each distribution center and its affiliated demanding points, use *Single-parent genetic algorithm* to solve.

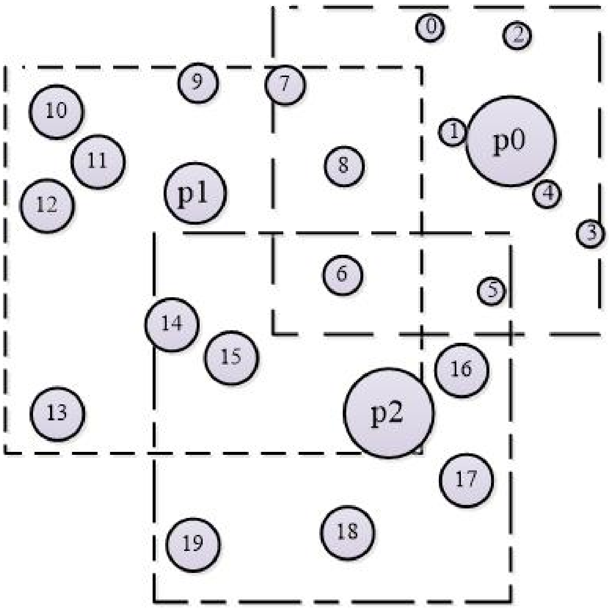

Figure 5: Set cover

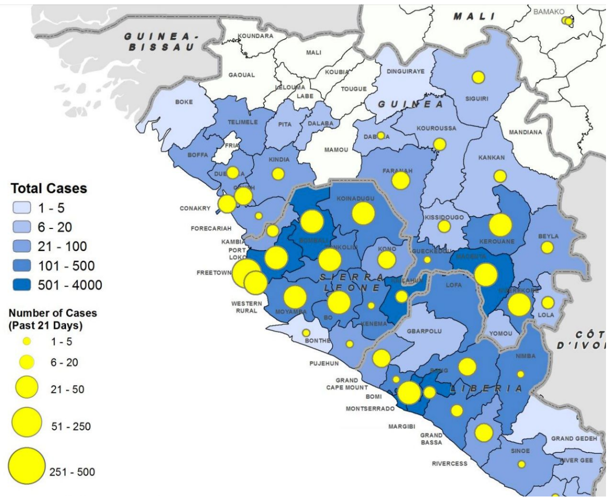

Figure 6: Africa demanding points

4.2 Sign definition

Scheduling model for emergency medicine distribution mainly includes five kinds of symbols, namely, sets, emergency supplies parameters, vehicle parameters, distance parameters, and decision variables.

To make this model clear, we use i,j to tag each point, especially, we use i=0 to represents the distribution center(that is, starting point).

- Sets:

  Set of demand points: $K = \{1, 2, \cdots, k\}$

  Set of emergency supply type: $\forall p \in P = \{1, 2\}$ (includes medicine and vaccine)

Set of vehicles: $\forall v \in V = \{1, 2, \cdots, n\}$

- Vehicle parameters:

  Q_v: The maximum load weight of vehicle v

  C_v: The fixed costs of vehicle v

  T_i: The arrival time at demand point i (if i=0,it means the departure time at supply point)

  S_{ijv}: The time needed from point i to point j for vehicle v, $S_{ijv} = \beta_{ij} \cdot d_{ij}$

  Z_{ijv}: The number of vehicles transporting items from point i to point j

- Distance parameters:

  d_{ij}: The distance from point i to point j

  α_{ij}: The economic weights of each unit length (transportation costs) from point i to point j for vehicle v

  β_{ij}: The timeliness weights of each unit length(Time consumption coefficient) from point i to point j for vehicle v

- Emergency supplies parameters:

  S_{iv}: The service time of unloading at point i

  R_{ip}: The demand for supply type p at point i

- Decision variables

  $$X_{ijv} = \begin{cases} 1\ , & \text{vehicles v move from i to j} \\ 0\ , & \text{otherwise} \end{cases}$$

  $$Y_{ipv} = \begin{cases} 1\ , & \text{vehicles v is in charge of transporting item i to point i} \\ 0\ , & \text{otherwise} \end{cases} \quad i = 1, 2, \cdots, k$$

- Demanding prediction:

  D_j: demand estimation for site j based on Demand prediction Model.

  To be more precise, we use triangular fuzzy number $\left(D_j^a, D_j^b, D_j^c\right)$.

  Their membership function is defined as:

$$\mu D_j = \begin{cases} 0, & D_j < D_j^a \\ \frac{D_j - D_j^a}{D_j^b - D_j^a}, & D_j^a \leqslant D_j < D_j^b \\ \frac{D_j^c - D_j}{D_j^c - D_j^b}, & D_j^b \leqslant D_j \leqslant D_j^c \\ 0, & D_j > D_j^c \end{cases}$$

  L_j represents the grade infection of region i, which ranges from 1 to 10 and infectious situation is more severe.

  δ_j represents the penalty factor of site j when its demand is not satisfied.

4.3 Mathematical Modeling

Distribution scheduling model aims at meeting the need of the demanding sites and shortening the distribution time as well as cost as soon as possible in order to curb the spread of disease and cure patients in time.

We define our objective function as

$$\min Z = \sum_{i=0}^{km}\sum_{j=0}^{km}\sum_{v=1}^{n} \alpha_{ij} \cdot d_{ij} \cdot X_{ijv} + \sum_{v=1}^{n} C_v \tag{5}$$

Constraints:

$$\sum_{p=1}^{m}\sum_{i=1}^{k} R_{ip} \cdot Y_{ipv},\ \ \forall v \in V,\ \ \forall p \in P; \Rightarrow \sum_{i=1}^{km} R_i \cdot Y_{iv} \leqslant Q_v,\ \ \forall v \in V \tag{6}$$

$$\sum_{v=1}^{n} Y_{ipv} = 1,\ \ \forall v \in V,\ \ \forall p \in P,\ \ i = 1, 2, \cdots, k;\ \ \Rightarrow \sum_{v=1}^{n} Y_{iv} = 1,\ \ \forall v \in V,\ \ i = 1, 2, \cdots, km; \tag{7}$$

$$\sum_{i=0}^{k} X_{ijv} = Y_{jpv},\ \ \forall v \in V,\ \ \forall p \in P,\ \ j = 0, 1, 2, \cdots, k;\ \ \Rightarrow \sum_{i=0}^{km} X_{ijv} = Y_{jv},\ \ \forall v \in V,\ \ j = 0, 1, 2, \cdots, km; \tag{8}$$

$$\sum_{j=0}^{k} X_{ijv} = Y_{jpv},\ \ \forall p \in P,\ \ i = 0, 1, 2, \cdots, k; \Rightarrow \sum_{j=0}^{km} X_{ijv} = Y_{jv},\ \ \forall v \in V,\ \ i = 0, 1, 2, \cdots, km; \tag{9}$$

$$\exists X_{ijv} = 1 : T_j = T_i + S_{iu} + S_{ijv} \ \ \Rightarrow \ \ T_j = X_{ijv} \cdot (T_i + S_{iu} + S_{ijv}),$$

$$i = 0, 1, \cdots, km, \quad j = 1, 2, \cdots, km, \quad i \neq j; \tag{10}$$

$$T_i \leqslant D_i \quad \Rightarrow \quad G(T_i) = \begin{cases} 0, & T_i \leqslant D_i \\ M, & T_i > D_i \end{cases} \tag{11}$$

$$Z_2 = \min \sum_{j} \left| E(D_j) - \sum_{i \in K}\sum_{j \in K}\sum_{r \in V} z_{ijr} Q_r \right| \delta_j L_j \tag{12}$$

The objective function is to make the distribution of emergency costs (total costs and fixed costs of vehicle use and) most minimized, which includes the vehicle transportation time from emergency supplies storehouses to demand point, and the supplies loading time at storehouses and unloading time at demand points.

Explanation:

Constraints (6) is loading capacity constraints, which should ensures the total demand for the infectious area each vehicle v is responsible for do not exceed its maximum capacity.

Constraints (8) shows that if the item p is transported to the demand point j by vehicle v, v is sure to arrive at j from the front point.

Constraints (9) shows that if the item type p is transported to the demand point j by vehicle v, v will head for the next point after unloading at j.

Constraints (10) is served time expression, reflecting the time association between affected spot j and the previous point owning the same car service.

Constraints (11) is defined as deadline constraints, which ensures the timeliness of medical distribution.

Constraints (12) is defined as the minimum risk and loss when the infectious site's need for medicine or vaccine is not satisfied. $E(D_j)$ means the expectation of triangular fuzzy number $\left(D_j^a, D_j^b, D_j^c\right)$, which could be calculated by defuzzification.

In order to apply model to algorithm to research further, we change direct constraints above in the form of penalty function and integrate to the target function,

$$\min Z = \sum_{i=0}^{km}\sum_{j=0}^{km}\sum_{v=1}^{n} \alpha_{ij} \cdot d_{ij} \cdot X_{ijv} + \sum_{v=1}^{n} C_v + M \cdot \max(\sum_{i=1}^{km} R_i \cdot Y_{iv} - Q_v, 0) + M \cdot \max(T_i - D_i, 0) \tag{13}$$

4.4 Optimization algorithm

A genetic algorithm (GA) is an artificial intelligence search method that is derived from the process of biological organism evolution. Genetic algorithms were described in University of Michigan 1960s and 1970s by John Holland. Genetic algorithms are a subset of the evolutionary algorithm set. The design of evolutionary algorithms is based on processes which occur in living organisms in nature, for example, inheritance, mutation, selection and crossover.

The GA requires a fitness statistics or function since it emulates the concept of solution evolution by stochastically developing generations of solution populations. This algorithm is applicable to search for the solution of high degree of complexity that often involves attributes that are large, non-linear and discrete in nature. GA can effectively and quickly solve multi-model, multi-target optimal function,which is key to our model, and therefore we apply GA to this medical distribution model.

Genetic algorithm process is shown in Fig.7.

4.4.1 Genetic Coding

To represent chromosomes in a genetic algorithm population, strings of bits are normally used, although there are many other existing methods currently in their infancy and

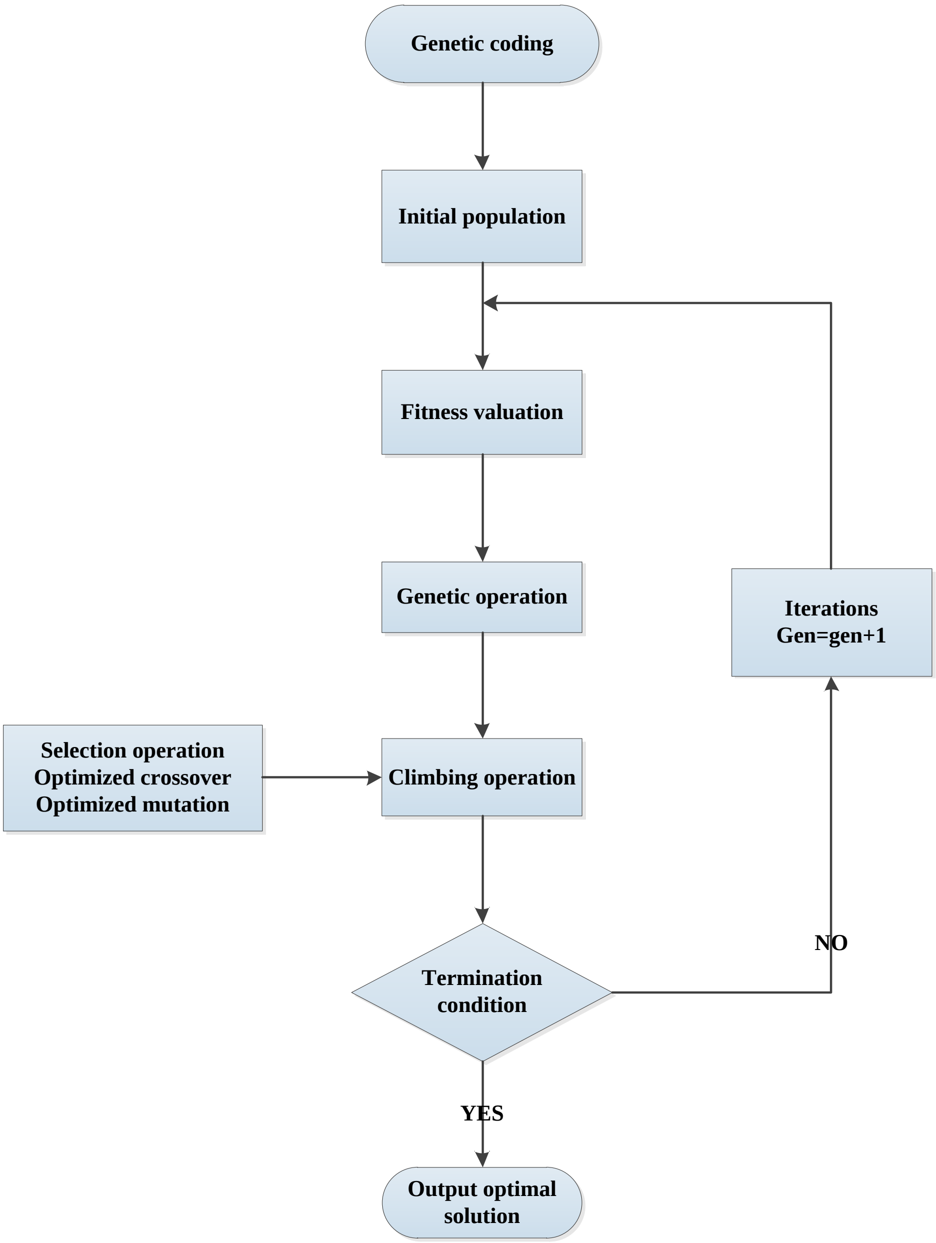

Figure 7: Genetic algorithm flow chart

under development such as fix-point binary representation, order-based representation, embedded list, variable element list and LISP expression. Every position (locus) in the string (chromosome) has two possible values (alleles): 0 and 1.

A solution(chromosome) is composed of several genes (variables). A chromosome represents a solution in the search space of potential solutions. A population contains a set of chromosomes which are composed of genes. In other words, the solution space contains a set of strings which are composed of bits.

The improved natural coding methods are adopt, a chromosome on behalf of a program of transport medicine emergency supplies. We use natural number coding to define a chromosome, 0 represents the distribution center (that is, the starting point), $\{1, 2, 3, \cdots, k\}$ represents the demanding sites.

To make our statement more precise, we explain the meaning of chromosome "01203450670". The vehicle assignment and route options is that three vehicles departure from the supply site and they distribute items to seven infectious area.

- The route for vehicle 1:

  supply site → infectious area 1 → infectious area 2 → supply site

- The route for vehicle 2:

  supply site → infectious area 3 → infectious area 4 → infectious area 5 → supply site

- The route for vehicle 3:

  supply site → infectious area 6 → infectious area 7 → supply site

4.4.2 Initial Population

The genetic algorithm belongs to groups based search method, we must prepare initial population of N individuals for genetic manipulation, each individual represents one solution. The initial group is the initial solution set randomly generated before evolutionary computing.

To improve convergence rate, we can generate initial group following these steps:

Step 1: sort the demand sites randomly

Step 2: Calculate from left to right, if the loading capacity of the first vehicle is greater than demand sum of the first k demand sites, and simultaneously less than demand sum of the first $(k + 1)$ demand sites, we will get the material needs substring $\{1, 2, 3, \cdots, k\}$ for which this vehicle is responsible.

Step 3: strike out the first k demand sites, calculate and define the material need substring for the next vehicle following the same steps. Do it repeatedly until all demand sites have been arranged.

Step 4: We insert 0 at the beginning and end of each substring. Finally, we connect these substrings to make up a complete chromosome.

Step 5: We reorder these demand sites, follow step 1-4 to gain another chromosome.

Repeat this process until the population size reaches N.

Fig.8 can demonstrate the relationship among population,chromosomes and genes visually.

4.4.3 Fitness Function

The principle here is "survival of the fittest". This operator chooses chromosomes in the population for reproduction. Based on an evaluation function, the "better" chromosomes

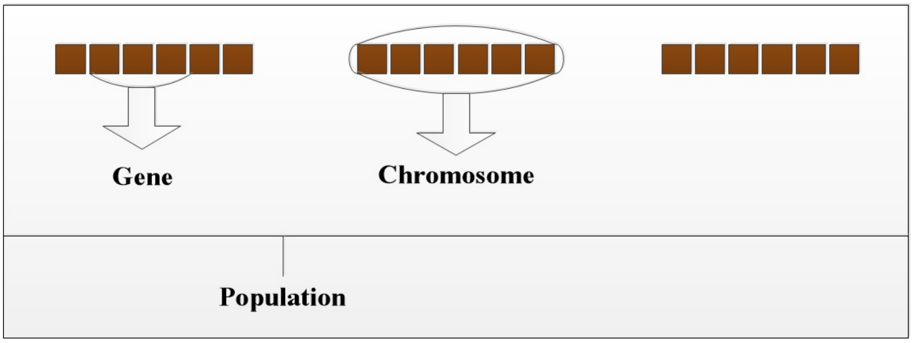

Figure 8: Population-Chromosomes-Genes

are identified and will be more likely to be selected for reproduction. Generally, the probability of selection is proportional to the fitness. It is possible that the best chromosome does not get selected in one run of the GA. However, if the GA is run many times, the probability of selection will converge towards its mathematical expected value.

This fitness operator's implementation is dependent on the specific problem at hand. As for this problem, we connect fitness function with our objective function Eqs.(13) to improve the result of model. In order to prevent the generation of supernormal fitness values and search retardation in the algorithm later, the nonlinear fitness function is employed to adapt to the evolution algebra dynamic adjustment:

$$f_h = \frac{Z_{min}}{Z_h}$$

In this formula, f_h represents fitness function result for chromosome h, Z_{min}represents the delivery cost of the best chromosome among the same generation, $Z_h$represents the delivery cost of the h-th chromosome.

The simulation results show that this fitness function can dynamically adjust to the individual fitness. Improved nonlinear fitness function, which is simple and easy to implement, can enlarge the differences between individuals and keep within group diversity.

4.4.4 Genetic Operation

Genetic operation is designed to simulate the nature biological evolvement. Its role is to achieve the survival of the fittest, and adapt to the environment and natural evolution.

(1) Selection

Selection is one of the key steps of genetic algorithm. It decides which individual can be passed down to next generation. In this algorithm, the selection operator uses nonlinear ranking selection. In nonlinear geometric ranking function, individual i is chosen at the probability computed complying with the following rules:

We combine classical French Roulette with retaining the best chromosome to choose function-well operator.

Step 1: Calculate fitness for every chromosome: $f_h, \ \ h = 1, 2, \cdots, n;$

Step 2: Sort chromosomes according to fitness, we choose the best fitness chromosome as the first individual of next generation:

Step 3: Calculate chosen possibility for every chromosome:

$$w_h = \frac{f_h}{\sum_{i=1}^{n} f_h}\ , \qquad h = 1, 2, \cdots, n;$$

Step 4: Calculate cumulative probability:

$$u_k = \sum_{i=1}^{k} w_k\ , \qquad k = 1, 2, \cdots, n;$$

Step 5: Generate uniformly distributed random numbers in $\left[0,1\right]$, if $r \leqslant u_l$, we will choose the first chromosome to generate the next generation; otherwise, we will choose the kth chromosoe as long as it meets:

$$u_{k-l} < r < u_k\ , \qquad k = 2, 3, \cdots, n$$

.

(2) Crossover

Crossover means the reorganization and exchange of two parents, aiming at generating two new individuals with higher adaptation degree value. This algorithm uses the single point crossover,arithmetical crossover and heuristic crossover.

As for our medical delivery model, we summarize two distinct features:

- sub-paths disorder among chromosomes
- Sub-path ordered within a chromosome

To make use of these features and retain better combination of parental genes, we organize crossover (Maximum retention Cross) as follows.

If both genes at intersections are 0, we use OX cross principle directly(the principle of cross based on sequence).

Otherwise, we move intersections left or right until reach the position where gene is 0,then we apply the OX principle.

Fig.9 and Fig.10 demonstrate this dynamic process vividly:

(3)Mutation

Crossover can transfer the fine genes of parent generation to next generation, which makes the child better than the parent. This algorithm takes the uniform mutation, combined boundary mutation and non-uniform mutation. This can prevent excellent genes in variability being damaged, but also introduce a new gene for the population on the local optimal solutions.

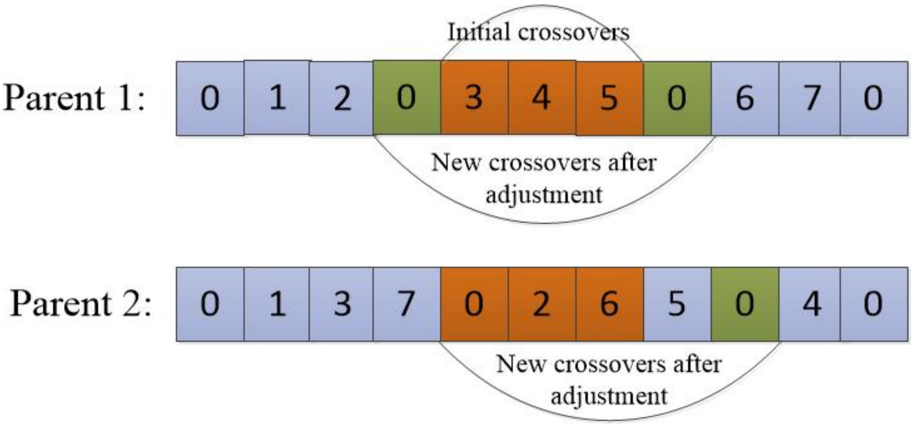

Figure 9: Previous-chromosomes

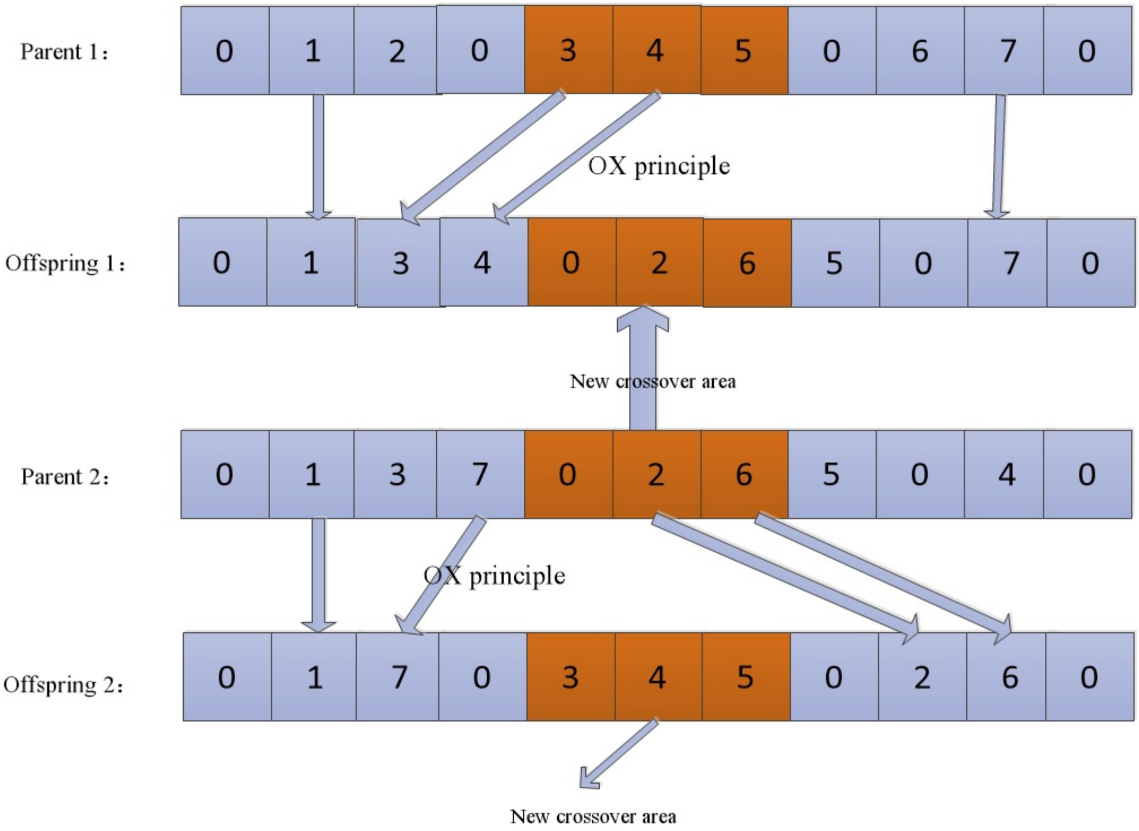

Figure 10: Latter-chromosomes

Fig.11 demonstrates our medicine distribution model visually.

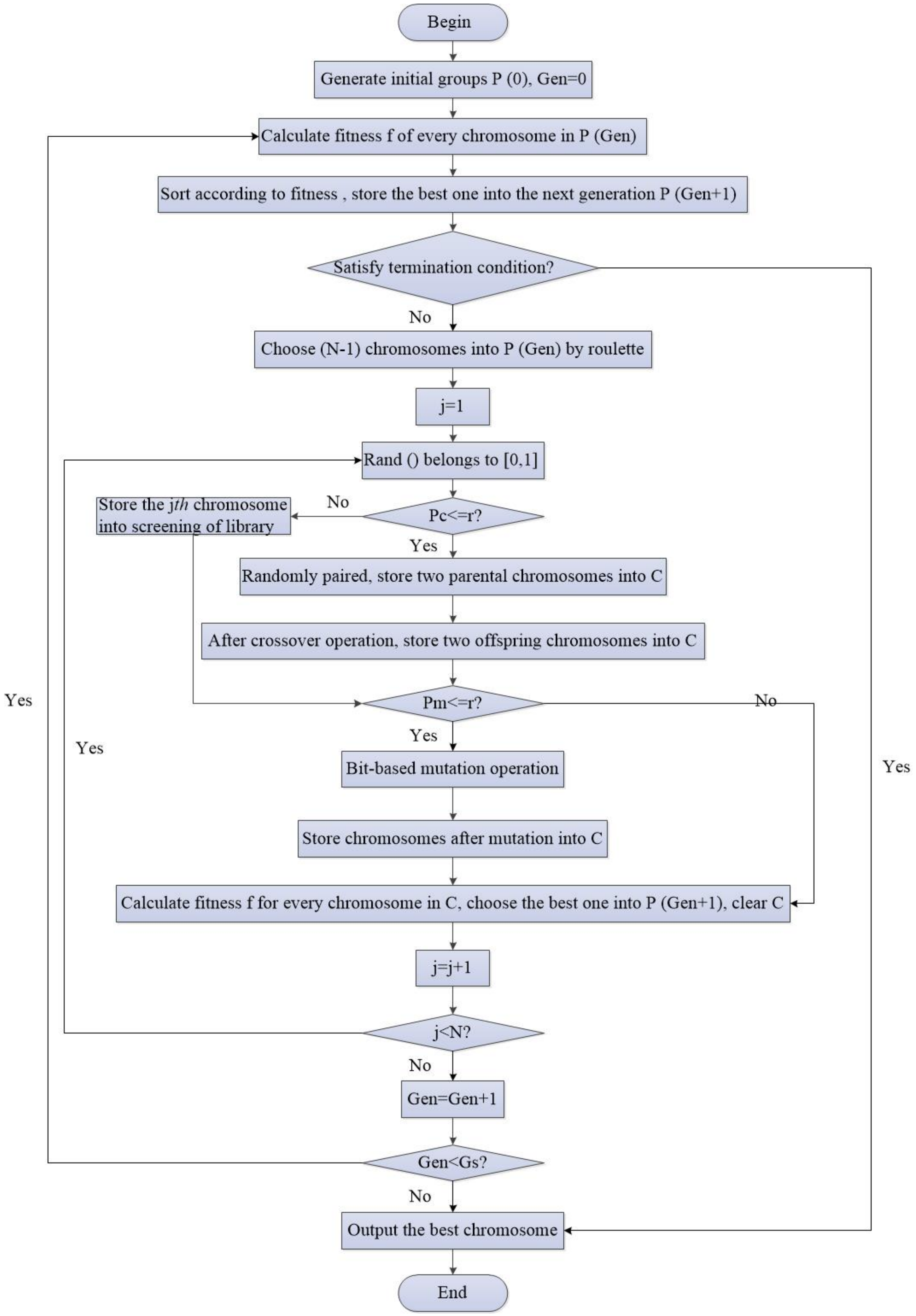
Begin
Generate initial groups P (0), Gen=0
Calculate fitness f of every chromosome in P (Gen)
Sort according to fitness , store the best one into the next generation P (Gen+1)
Satisfy termination condition?
No
Choose (N-1) chromosomes into P (Gen) by roulette
j=1
Rand () belongs to [0,1]
Pc<=r?
No
Store the jth chromosome into screening of library
Yes
Randomly paired, store two parental chromosomes into C
After crossover operation, store two offspring chromosomes into C
Pm<=r?
No
Yes
Bit-based mutation operation
Store chromosomes after mutation into C
Calculate fitness f for every chromosome in C, choose the best one into P (Gen+1), clear C
j=j+1
j<N?
No
Gen=Gen+1
Gen<Gs?
No
Output the best chromosome
End
Yes
Yes
Yes

Figure 11: GA distribution simulation

4.5 Example Analysis

As we stated before, we define point 0 as supply point. Now we apply genetic algorithm to solving two type items-medicine and vaccine delivery.

4.5.1 Simulation assumption

Table 5 and Table 6 show cost matrix and time-consuming matrix. Time limits and unloading time are also demonstrated in Table 8 and Table 9. To simplify our model, we assume:

(1)The maximum capacity for every vehicle is 200;

(2)The fixed cost for every vehicle is 100;

The initial population is 200, crossover probability is 0.85, mutation probability is 0.05, the max generation is 1000.

From previous Demand Forecast model, we have got the demand for these twenties sites. To simplify our distribution model, we choose eight sites 1-th demand for medicine and vaccine from previous results.They are shown clearly in Table 7.

We make use of genetic algorithm to arrange vehicles and routes to optimize our objective function.

Table 5: Inter-node cost matrix based on distance

Point	Area-A	Area-B	Area-C	Area-D	Area-E	Area-F	Area-G	Area-H
O	40	39	11	65	47	84	26	53
A		25	38	22	51	44	30	49
B			64	41	30	27	75	13
C				36	82	50	67	18
D					23	58	94	39
E						72	54	45
F							18	66
G								51

Table 6: Inter-node time-consuming matrix based on distance

Point	Area-A	Area-B	Area-C	Area-D	Area-E	Area-F	Area-G	Area-H
O	1.3	3	0.4	2	0.8	1.5	1.2	2
A		1.6	2.1	1.2	1.3	2.6	1.2	2.4
B			0.7	1	1.3	1.5	2.5	1.1
C				1	2.7	1.4	2.3	0.5
D					2.5	1.7	0.8	1.2
E						1.5	2	2.2
F							0.6	1
G								2.1

4.5.2 Simulation Result

The best fitness scored chromosome is 0-6-7-18-19-20-0-8-9-4-5-0-1-2-3-0-11-12-13-10-0-14-15-16-17-0.

This chromosome stands for the distribution strategy shown in Table 10.

Due to the complexity of real world problem, we simplify the model and construct the example for verify the effectiveness of our model and the feasibility of our algorithm. Here we can see our average load is approximately equal to 85%, while the arriving time is a little ahead of the deadline. Thus, we draw conclusions that we should meet the time require first. And after time require is satisfied, we should save the cost in distribution. One effective way is sending different kind of supplies of one area by single vehicle as long as possible.

4.5.3 Further discussion

When we are conducting 797th genetic manipulation, the result meets algorithm termination condition. In Fig.12, the horizontal axis represents generation of computing, the vertical axis represents function value of H. From function image, the curve above represents average of H for every generation, from 53579(the first generation) converging to 14368. The curve below represents average of F for every generation, from 14238(the first generation) converging to 12670.

From function image Fig.13, the horizontal axis represents generation of computing,

Table 7: The demand for vaccine and medicine

Area	Area-A	Area-B	Area-C	Area-D	Area-E	Area-F	Area-G	Area-H
vaccine	112	99	81	77	210	98	76	252
medicine	20	82	39	98	48	85	96	55

Table 8: Deadline for items

Point Type	Area-A	Area-B	Area-C	Area-D	Area-E	Area-F	Area-G	Area-H
medicine	10:00	12:00	15:00	11:00	10:00	13:00	9:00	12:00
vaccine	18:00	16:00	18:00	16:00	12:00	17:00	15:00	14:00

Table 9: Unloading time(min)

Point Type	Area-A	Area-B	Area-C	Area-D	Area-E	Area-F	Area-G	Area-H
medicine	28	25	10	18	52	24	19	63
vaccine	10	41	21	49	24	36	44	28

Table 10: Unloading time(min)

Vehicle	Capacity	Delivery order	Actual distribution line	Arriving time	Punish cost	Rescue cost
1	140	5,6,13,14,16	O-C(I)/C(II)-G(I)/G(II)-O	7:00-8:24-9:04-10:00	0	388
1	200	7,8,13,3,4	O-D(I)/D(II)-B(I)/B(II)-O	7:00-8:44-9:24-10:30	0	458
1	180	9,10	O-E(I)/E(II)-O	7:00-8:04-8:36-9:00	0	280
1	190	11,12	O-F(I)/F(II)-A(I)/A(II)-O	7:00-8:30-8:56-10:20-	0	386
1	140	19,20	O-H(I)/H(II)-O	7:00-8:30-9:36-10:18-	0	444

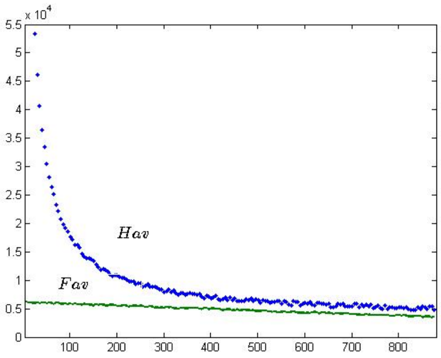

Figure 12: Means of H and F in each generation

the vertical axis represents function value of minH and minF. The curve above represents the minimum value of the objective function H, from 42363(the first generation) converging to 13434. The curve below represents the minimum value of the objective function F, from 11830(the first generation) converging to 10745. We can arrive at conclusion from that: When we are conducting 568th genetic manipulation, the minimum value of the objective function decrease slowly.

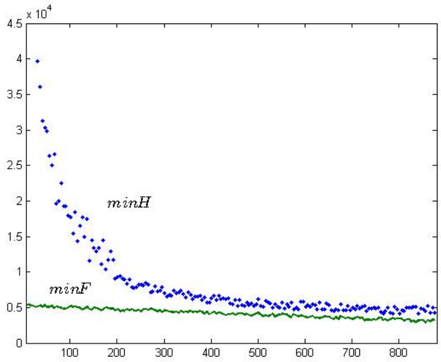

Figure 13: Least H and least Fin each generation

From function image Fig.14,The horizontal axis represents generation of computing, the vertical axis represents function value of punishment, from 14917(the first generation) converging to 637.

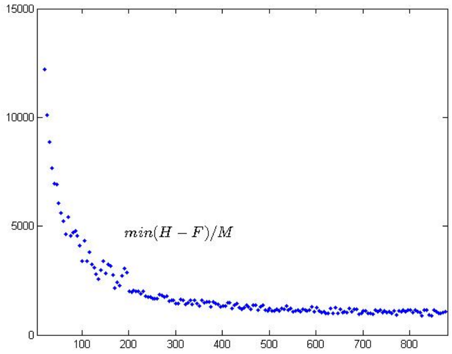

Figure 14: Least punishment in each generation

Punishment mechanism, When solution space has none feasible solution, give adaption degree a punishment mechanism to decrease the adaption degree of this individual. The chance of its gene passing to the next generation will decrease. We use this formula to regulate adaption degree:

$$F'(X) = \begin{cases} F(X) & (when X) \\ F(X) - P(X) & (when X) \end{cases}$$

F(X): initial adaption degree, FâĂŹ(X): new adaption degree under punishment mechanism, P(X): punishment function. The final objective function is:

$$\min \sum_{t\in T}\sum_{d\in D}\sum_{a\in A}(\lambda_{tda}\cdot diss_{tda} - \sum_{j=1}^{k_a}\gamma_{t,t+j,da}\cdot com_{t,t+j,da}) + M\sum_{t\in T}\sum_{d\in D}\sum_{a\in A}\max(X_{tsa} - sur_{t-1,sa} - m_{tsa}, 0)$$

- λ_{tda}: the loss caused by drugs a at point d not fulfilling in period t.
- X_{tsa}: the expected number of drugs a taken from point s in period t.
- Y_{tda}: the expected number of drugs a brought to point d in period t.
- sat_{tda}:the expected number of drugs a fulfilling at point d in period t.
- $diss_{tda}$:the expected number of drug a not fulfilling at point d in period t.
- $com_{t,t+j,da}$:the expected number of drug a not fulfilling at point d in period t but fulfilling in period t+j.
- $sur_{t-1,da}$:the expected number of drugs a left at point d in the end of period t.
- $sur_{t-1,sa}$: the expected number of drugs a left at point s in the end of period t.
- Objective function: in all time, all point, all drugs xmum loss.
- $\gamma_{t,t+j,da}$: At point d, the reduction of loss when per drug a not fulfilling at point d in period t but fulfilling in period t+j.

5 Sensitivity Analysis

We test our model in both large population size and small population size for various changes in our assumptions. At the crossover rations 0.85, mutation rations 0.05, We test different population size for 200,205,210,215,220, respectively.

The data table of our experiment result is shown in Table 11. ($Hmin$ stands for the minimum punish function including score existed. $Fmin$ stands for minimum punish function excluding score existed.)

We have From the data, we conclude that the larger size of population, the richer chromosome diversity of the group, the higher quality the solution we find. However, the number of generate manipulation grow simultaneously. And the time cost each iteration increase. if $N > 200$, the quality improvement is not obvious, but the time cost increase more faster. Consider the time cost, we will apply $N = 200$ at the next steps.

Table 11: Output data at different group size

N	200	205	210	215	220
$Hmin$	12306	12210	12087	11954	11940
$Fmin$	9902	9675	9702	9303	9505
$Timecost$	15	16	18	20	23

In this section, we will test two critical parameters: crossover ratios P_c and mutation ratiosP_m.

Based on the condition $N = 200$ and Mutation ratio $P_m = 0.05$,we test different crossover ratios, 0.75,0.8,0.85,0.9,0.95. The output data shown in Table 12.

We can make conclusion from the experiment data that, at the condition $P_c < 0.85$, the greater crossover ratio is, the faster we can find a approximate optimal solution. Convergence rate become faster and we can get solution in less number of iterations. So the calculating time cost is reduce correspondingly. While $P_c > 0.85$, the global optimization's characters will not maintain long to pass on to the next generation. So it will be hard to get a more optimal solution, no matter how many generation we iterate.

Table 12: Output data at different crossover ratios

P_b	0.75	0.8	0.85	0.9	0.95
$Hmin$	12316	12310	12306	12312	12320
$\frac{\Delta H_i}{H_{0.85}}$	7.3×10^{-4}	2.4×10^{-4}	0	4.0×10^{-4}	1.05×10^{-3}
$Fmin$	10226	10097	9902	10188	10005
$\frac{\Delta F_i}{F_{0.85}}$	3.3×10^{-2}	2.0×10^{-2}	0	2.9×10^{-2}	1.0×10^{-2}

At the condition of $N = 200$,$P_c = 0.85$,we test different crossover ratios, 0.05,0.06,0.07,0.08,0.09. The output data shown in Table 13.

We can make conclusion from the experiment data that, at the condition $P_m < 0.05$, the greater mutation ratio is, the faster we can find a approximate optimal solution. Convergence rate become faster and we can get solution in less number of iterations. So the calculating time cost is reduce correspondingly.

While $P_m > 0.05$, the global optimization's characters is probably mutate to other characters. So it will be hard to get a more optimal solution, it could cost a longer time.

Table 13: Output data at different mutation ratios

P_b	0.05	0.06	0.07	0.08	0.09
$Hmin$	12306	12287	12274	12266	12259
$\frac{\Delta H_i}{H_{0.85}}$	0	-1.5×10^{-3}	-2.6×10^{-3}	-3.3×10^{-3}	-3.8×10^{-3}
$Fmin$	9902	9886	9874	9868	9860
$\frac{\Delta F_i}{F_{0.85}}$	0	-1.6×10^{-3}	-2.8×10^{-3}	-1.5×10^{-3}	-3.4×10^{-3}

6 Conclusions

In this paper,we studied optimal combination of vaccination and treatment strategies for driving Ebola with cure and vaccine towards eradication within a specified period. We considered an SIR model with varying size population using vaccination and treatment as control measures. Numerical simulations showed that the resulting optimality system could result in a rapid reduction in the susceptible numbers as well as an appreciable reduction in the infectious number.

Simultaneously,the optimization model is established with an aim to minimize the total transportation time and satisfy every demand point as much as possible, and genetic algorithm is used to solve the model. The process result shows that the model and its algorithm can meet the demand of medical delivery scheduling optimization for medicine and vaccine distribution, and the obtained solution shows good performance in some aspects. Our stimulation results prove our model reduces the delivery time and avoids premature convergence of the algorithm. As a result, the model and its algorithm are suitable for Ebola eradicating problems.

7 Strengths and weaknesses

7.1 Strengths

- Our sensitivity analyses show that our models are fairly robust to changes in parameter value.
- The demand forecast based on classical SIR model uses bilinear incidence while the objective functional minimizes the number of infectious people at the end of the control period (terminal time) to a level-herd immunity level, at which the disease can naturally die out without any subsequent intervention.
- We deal with the optimal control of the disease and consider a variable population size (tending to a limit) varying with our medical control strategy, which is more realistic and convenient for us to establish distribution system.
- The category of emergency medicine supplies we considered include treatment drug as well as vaccine, which makes the whole transport network closer to real situation.
- It can be seen from the convergence curve, with the basic genetic algorithm, the improved genetic algorithm containing penalty function items, convergence rate is slower, but it gets better solution and has been less distribution time. To a certain extent, it also avoids early shortcomings of the genetic algorithm.
- We come up with various criteria to compare different situations. Our model is based on constraints in many aspects.

7.2 Weaknesses

- Some of the parameters are based on semi-educated guesses because few data are available. However, based on our sensitivity analysis, they will not make a great difference if slightly changed.
- Factors of human judgments may be over-simplified. To consider that mobility in different areas, the actual situation about the spread of Ebola may be more complicated.
- Due to the complexity of real world problem,genetic algorithm may give local optima solution,but it has optimize this problem to a rather large extent compared with initial solution.

References

[1] Anwen Lu,Xiangjun Xie, "Logistics and distribution model under emergency situation", Southeast Petroleum Institute, vol, 14, no.8, pp.23-25, 2003.

[2] Jin Li, etc, "Multi-resource emergency scheduling model and algorithm in disaster chain", System Engineering Theory and Practice, vol.7, no.3, pp.106-113, 2011.

[3] Jun Tian, etc, "Emergency supplies distributing and vehicle routes programming based on particle swarm optimization", systems Engineering Theory and Practice, vol. 14, no. 8, pp.25-29, 2011.

[4] Sen Chen, etc, "Model for emergency relief distribution using delay-resource conversion", Computer Engineering and Applications, vol. 15, no. 8, pp. 215-221, 2011.

[5] S. Lenhart, J. Wortman. Optimal control applied to biological models. Taylor & Francis, Boca Raton, 2007.

[6] S. Nanda, H. Moore, S. Lenhart. Optimal control of treatment in a mathematical model of chronic myelogenous leukemia. Mathematical Biosciences, 2007, 210: 143-156.

[7] Xiling Hu, etc, "Path planning based on adaptive chaotic genetic algorithm", Computer Engineering and Applications, vol. 15, no. 1, pp. 75-80, 2012.

[8] Biao Luo, Jinhua Zheng, "genetic algoyithm based on the directed climbing", Computer Engineering and Applications, vol. 44, no. 6, pp. 92-95, 2008.

[9] Jin Qin, Optimization models and algorithms for multicommodity logistics network design, PhD Dissertation, Central South University, Changsha, China, 2006.

Appendices

Appendix A Announcement letter

Since Ebola outbreak in West Africa was first reported in March 2014, 9,019 people has been reported as having died from the disease in six countries. It has rapidly become the deadliest occurrence of the disease since its discovery in 1976. And the number of people infected Ebola is increasing rapidly. Weekly case incidence increased in all three countries for the first time this year. Ebola epidemic is the most serious disease all over the world and it threatens everybody's life.

Studying day and night, experts eventually develop medication for Ebola. Through several times of tests, it works well in treatment. This is our strong weapon to defeat Ebola.

Not only have that, our association have made detailed investigation and analysis on the affected areas and the situation. With huge amount of data and analysis, we now promote a full plan to help solve the problem. Our aim is helping eradicating the Ebola virus by forecasting every areas' demand and optimize the medicine distribution system.

We can forecast the medicine demand according to the composition of local people, the partition shared by susceptible, infectious and normal people. The analysis system is based on the data that hospital and medical department posted weekly and then give out our prediction for the next stage. Since medicine distribution center specifically for the Ebola virus is to be established. Our goal is to generate a delivery strategy to minimize the cost, which balance the time cost, transportation cost, fixed cost, product cost, etc. The distribution strategy will be different depending on the severity and the distribution of the epidemic areas. With this technique, we guarantee that will we delivery both vaccine and medical treatment to Ebola-infected people as quickly as possible.

In the past 40 years, the whole world could do nothing to Ebola. Especially in the last year, the merciless virus took away thousands of lives. However, we are about to bid farewell to this painful memories, completely eradicating Ebola, making the most effective protection for the health of people around the world.

The establishment of drug production, distribution centers, leasing vehicle, and communication between countries are undertaken. The victory of the battle will be just around the corner. Take out of faith and courage to overcome the disease!

Appendix B Appendix Code

Input Java source:

```java
package com.genetic;
import java.util.ArrayList;
import com.display.LineFrame;
public class Group
{
    ArrayList<Member> curgenerate=new ArrayList<Member>();
    ArrayList<Memberplus> roulette=new ArrayList<Memberplus>();
    int  average;
    int count=0;
    int e,f;
    LineFrame lf = new LineFrame();
    public Group()
    {
        for(int i=1;i<=Base.memberSize;i++)
        {
            curgenerate.add(new Member(Base.RandomGenerate()));
        }
        CalScore();
    }
    public void evolution()
    {
        int num=0;
        while(num<Base.iterations)
        {
            nextGenerate();
            num++;
            CalScore();
        }
    }
    private void nextGenerate()
    {
        curgenerate.clear();
        while(curgenerate.size()<Base.memberSize)
        {
            Member parent[]={getRandomparent(),getRandomparent()};
            while(parent[0]==parent[1])
                parent[1]=getRandomparent();
            System.out.println("father0= "+parent[0].dna+" "+parent[0].getScore());
            System.out.println("father1= "+parent[1].dna+" "+parent[1].getScore());
            Member  son[]=mating(parent);
            curgenerate.add(son[0]);
            curgenerate.add(son[1]);
            System.out.println("son0= "+son[0].dna+" "+son[0].getScore());
            System.out.println("son1= "+son[1].dna+" "+son[1].getScore());
        }
    }

    private Member getRandomparent()
    {
        int point=(int)(Math.random()*count);
        for(Memberplus n:roulette)
        {
            if(n.begin<=point&&point<=n.end)
                return n;
        }
        return null;
    }

    private class Memberplus extends Member
    {
        private int begin,end;
        public Memberplus(Member m,int b,int e) {
            super(m);
            begin=b;
            end=e;
        }
    }
    void CalScore()
    {
        count=0;
        roulette.clear();
        boolean flag=false;
        flag=false;
        for(Member n:curgenerate)
        {
            int tmp=n.getScore();
            tmp=1500-tmp;

            if(!flag)
            {
                e=tmp;
                f=tmp;
                flag=true;
            }
            if(tmp>e)
            {
                e=tmp;
            }
            if(tmp<f)
            {
                f=tmp;
            }
            roulette.add(new Memberplus(n, count, count+tmp-1));
            count+=tmp;
        }
        average=count/Base.memberSize;
        lf.addPoint(e*Base.enheight, f*Base.enheight, average*Base.enheight);
    }

    int Check(ArrayList<Integer> dna,int st,int en)
    {
        for(int i=st+1;i<en;i++)
        {
            if(dna.get(i)==0)return 1;
        }
        return 0;
    }

    public ArrayList<Integer> Variation(ArrayList<Integer> dna)
    {
        int st=(int)Math.random()*dna.size();
        int en=(int)Math.random()*dna.size();
        if(st>en)
        {
            int tmp=st;
            st=en;
            en=tmp;
        }
        if(Check(dna,st,en)==1)
        {
            int tmp=dna.get(st);
            dna.set(st,dna.get(en));
            dna.set(en,tmp);
        }
        return dna;
    }

    public Member[] mating(Member[] parents)
    {
        if(Math.random()<0.7)
        {
            int exist[]=new int[Base.pointNumber+1];
            ArrayList<Integer> dna0=new ArrayList<Integer>();
            int st=(int) (Math.random()*parents[0].dna.size());
            int en=(int) (Math.random()*parents[0].dna.size());
            if(st>en)
            {
                int tmp=st;
                st=en;
                en=tmp;
            }
            while(parents[0].dna.get(st)!=0)
            {
                st=st-1;
            }
            while(parents[0].dna.get(en)!=0)
            {
                en=en+1;
            }
            System.out.println("st1 en1 "+st+" "+en);
            for(int i=st;i<=en;i++)
            {
                dna0.add(parents[0].dna.get(i));
                exist[parents[0].dna.get(i)]=1;
            }
            int p=0;
            int left=Base.Qv;
            while(p<parents[1].dna.size())
            {
                if(parents[1].dna.get(p)>0&&exist[parents[1].dna.get(p)]==0)
                {
                    exist[parents[1].dna.get(p)]=1;
                    if(left-Base.baseGraph.get(parents[1].dna.get(p)-1).getVal()<0)
                    {
                        left=left+Base.Qv-Base.baseGraph.get(parents[1].dna.get(p)-1).getVal();
                        dna0.add(0);
                        dna0.add(parents[1].dna.get(p));
                    }
                    else {
                        left-=Base.baseGraph.get(parents[1].dna.get(p)-1).getVal();
                        dna0.add(parents[1].dna.get(p));
                    }
                }
                p++;
            }
            if(dna0.get(dna0.size()-1)!=0)
                dna0.add(0);
            exist=new int[Base.pointNumber+1];

            ArrayList<Integer> dna1=new ArrayList<Integer>();
            st=(int) (Math.random()*parents[1].dna.size());
            en=(int) (Math.random()*parents[1].dna.size());
            if(st>en)
            {
                int tmp=st;
                st=en;
                en=tmp;
            }
            while(parents[1].dna.get(st)!=0)
            {
                st=st-1;
            }
            while(parents[1].dna.get(en)!=0)
            {
                en=en+1;
            }
            for(int i=st;i<=en;i++)
            {
                dna1.add(parents[1].dna.get(i));
                exist[parents[1].dna.get(i)]=1;
            }
            p=0;
            left=Base.Qv;
            while(p<parents[0].dna.size())
            {
                if(parents[0].dna.get(p)>0&&exist[parents[0].dna.get(p)]==0)
                {
                    exist[parents[0].dna.get(p)]=1;
                    if(left-Base.baseGraph.get(parents[0].dna.get(p)-1).getVal()<0)
                    {
                        left=Base.Qv-Base.baseGraph.get(parents[0].dna.get(p)-1).getVal();
                        dna1.add(0);
                        dna1.add(parents[0].dna.get(p));
                    }
                    else {
                        dna1.add(parents[0].dna.get(p));
                        left-=Base.baseGraph.get(parents[0].dna.get(p)-1).getVal();
                    }
                }
                p++;
            }
            if(dna1.get(dna1.size()-1)!=0)
                dna1.add(0);

            if(Math.random()<0.1)
            {
                dna0=Variation(dna0);
                dna1=Variation(dna1);
            }

            Member res[]={new Member(dna0),new Member(dna1)};
            return res;
        }
        else
        {
            ArrayList<Integer> dna0=parents[0].dna;
            ArrayList<Integer> dna1=parents[1].dna;
            if(Math.random()<0.1)
            {
                dna0=Variation(dna0);
                dna1=Variation(dna1);
            }

            Member res[]={new Member(dna0),new Member(dna1)};
            return res;
        }
    }
    public int getAverage() {
        return average;
    }
}
```