



2019 年湖南省高校第四届研究生数学建模竞赛参赛承诺书

我们仔细阅读了湖南省高校研究生数学建模竞赛的竞赛规则。

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛规则的，如果引用别人的成果或其他公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们郑重承诺，严格遵守竞赛规则，以保证竞赛的公正、公平性。如有违反竞赛规则的行为，我们将受到严肃处理。

我们授权湖南省高校研究生数学建模竞赛组委会，可将我们的论文以任何形式进行公开展示（包括进行网上公示，在书籍、期刊和其他媒体进行正式或非正式发表等）。

我们参赛选择的题号是（从组委会提供的试题中选择一项填写）：

我们的参赛报名号为（如果组委会设置报名号的话）：

所属学校（请填写完整的全名）：

参赛队员（打印并签名）： 1.

2.

3.

指导教师或指导教师组负责人（打印并签名）：

日期： 年 月 日

评阅编号（由组委会评阅前进行编号）：

2019 年湖南省高校第四届研究生数学建模竞赛

编号专用页

评阅编号（由组委会评阅前进行编号）：

评阅记录（可供评阅时使用）：

[illegible]



2019 年湖南省高校第四届研究生数学建模竞赛

题 目： 猪场连续生产销售计划

摘 要：

本模型首先将题中所给的约束条件以及各项收入和支出进行抽象和建模，并通过逻辑推理一步步简化约束条件，首先计算出整个连续生产计划中的部分最优解，然后在该部分解的基础上建立时间推进的系统仿真模型，通过遗传算法求解生产计划的另一部分解，最后得到一个符合约束条件的完整的最优连续生产计划。

针对配种和更新计划的制订，本文分析出影响配种计划和生产猪更新计划的最大因素即为 2019 年预测销售计划，通过预测销售计划以及实际猪只生产规律，可进行逆推得到最理想的配种计划，并作出最小代价的调整来满足其他约束条件。而更新计划的制定原则是调整生产母猪内部各种类的比例，平衡各类母猪的生产负担。另外，本文考虑尽可能减少更新生产猪的代价，即尽可能使更新母猪都是自留产生的，可以得到加入自留母猪生产的总的配种计划。

针对副产品的销售计划，本文首先运用问题一中得到的解来构建一个没有副产品销售的时间推进仿真系统，包括了所有猪只在任何时间节点的状态，而将制定的副产品销售计划输入仿真系统中即可得到副产品利润，以此作为目标函数，在满足其他约束条件的前提下尽可能使其最大。本文选择遗传算法对问题进行求解，最后得到使目标函数最大的最优副产品销售计划。

针对饲料消耗量测算问题，根据前两个问题求得的解，结合题中所给信息，可以得到一个完整的猪场连续仿真系统，从而能够对每周各饲料消耗量进行求解。

关键词：遗传算法，离散仿真，约束优化，任务规划

一、问题重述与分析

本问题的主要任务即在给定的某养猪场的初始状态以及各种约束条件，制定一个猪场连续生产销售计划。该猪场的初始状态即为初始猪只存栏，包含了该猪场所有猪的必要信息和状态。该问题的主要聚焦点是各种约束条件，包括猪只生产规律、猪舍容量、运输能力等，其中一个隐藏的约束条件是指定的 2019 年各品种猪的月销售计划。目标是实现猪场连续生产，在给定猪场收益的具体组成和计算方法的情况下，达到收益的最大化。

重点工作是：1) 制定每周配种计划以及生产猪的更新计划，2) 制定副产品销售计划以及 3) 测算每周各类饲料的消耗量。问题三的结果由前两个问题的解决定，因此重点分析的是配种计划、更新计划以及副产品销售计划的制定。而将这三个计划作为一个整体的解来求将极大地增加工作难度，故本模型将两个问题分开进行考虑。

针对问题一，即配种计划和更新计划的制订，本模型最优先考虑的是 2019 年预测销售计划，尽量使各类种猪的生产量精确满足销售计划，可以通过逆推的方法得到一个最理想的配种计划，在这个计划的基础上改变部分值以满足产房容量等直观约束条件。这部分内容的主要思考点就是对理想配种和更新计划做出最小的调整；

针对问题二，即副产品销售计划的制订，根据问题一中已经求出来的解可以确定一个没有副产品销售的系统仿真模型，问题二在此基础上变成了一个比较经典的优化问题，通过分析约束条件、目标函数以及运用合理的优化方法可以对该问题进行求解；

问题三，饲料消耗量测算，则是在前两问求得解的基础上做出的运算。

二、假设

1. 假设公猪母猪的生长数据和生长模式没有区别；
2. 假设每年的销售计划与 19 年的销售计划一致，以取得题意中**连续**生产的收益最大化和可持续化；
3. 假设现有的生产母猪均没有受孕，且可以立即生产；
4. 假设生产公猪只能从外面，不存在自留，且买来的生产公猪和生产母猪能够马上投入使用；
5. 假设该猪场只售卖销售计划包括的猪的种类；
6. 假设种猪完成选种后即可被售卖；
7. 题目信息与附件信息存在出入时，以题目中信息为准；譬如题意中“分娩哺乳期 21 天”，但是在附件 6-猪只存栏中部分状态为“哺乳仔猪”的日龄超过 21 天（最高 28 天）；

三、符号说明

符号	备注
$R_{主}$	主产品营业额
$R_{副}$	副产品营业额
$C_{买猪}$	买猪成本
$C_{运输}$	运输成本
$C_{饲养}$	饲养成本
r_i	事件 <i>i</i> 发生的概率
N_{type}	类型为 type 的母猪产活仔数 type∈ [A, B, C, D]
u_{ij}	第 j 周向第 i 周借的种猪的数量
x_{ij}	第 j 周 i 类母猪的配种数量
M_{ij}	第 j 周 i 类母猪的需求量
Y^1	所有副产品猪的售卖顺序
Y^2	每周售卖的猪的数量

四、配种计划和更新计划模型

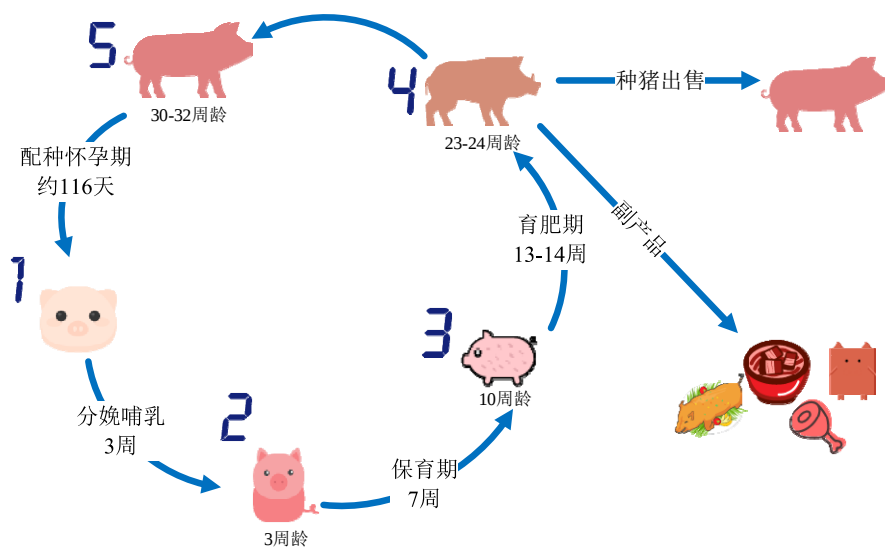


图 1 猪的生长周期图示

为了方便对问题的分析和阐述，根据题意中的阐述，作出图 1 的生长周期

图。

4.1 问题的抽象与建模

对于该问题需要达到的目标，即实现连续生产的收益最大化，首先需要进行具体的建模。该猪场的主产品是种猪，其他的则为副产品，因此本文考虑利润最大化目标时，首先考虑的是主产品的销售，即种猪产量是否符合销售计划。

整个问题可以看成是一个复杂的目标约束规划问题，其中，
未知变量为每周或每日的配种计划、猪场生产猪的更新计划以及副产品销售计划，主要以矩阵的形式存储展示；

约束条件包括问题中所给的猪只生产规律约束、生产猪更新约束、猪舍容量约束、运输能力约束以及销售计划约束；

而**目标函数**则是利润函数，为主产品销售额、副产品销售额减去运输成本、买猪成本、饲养成本。

其具体数学形式为

$$\min L = R_{\text{主}} + R_{\text{副}} - C_{\text{买猪}} - C_{\text{运输}} - C_{\text{饲养}}$$

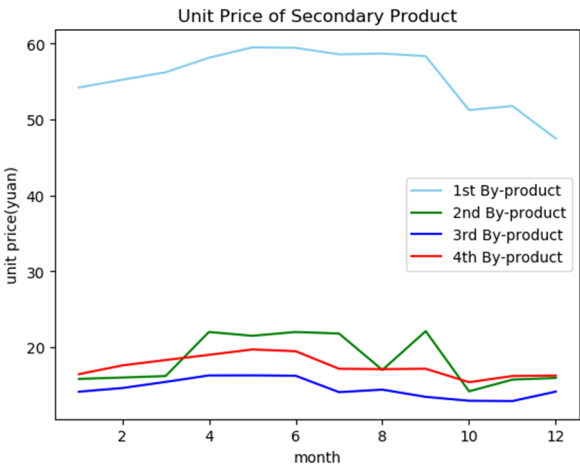


图 2 副产品价格月份走势图

由上图 2 可知，除了副产品一/二在 9 月份之后、以及副产品二在 3 月/8 月价格跳水，不同副产品的单位价格走势趋势基本一致。

对于类似复杂的优化问题，无法用常规的解法来对其进行求解，因此本模型考虑从以下两个方面入手，来确定猪场配种、更新计划以及副产品销售计划：

- 从 2019 年预测销售计划、生产猪更新率以及产房约束得到配种计划以及更新计划；
- 由配种计划和生产猪的更新计划确定每个时间节点的养猪场状态，在满足猪舍容量约束、运输能力约束的条件下确定副产品销售方案，使副产品销售利润最大。

4.2 生长模型设计

由于在本模型中需要对猪进行配种、产仔等，并对猪场的副产品销售进行预测与规划，这需要模型根据猪的当前日龄预测该猪当前的体重。

附件 2 中给出了部分猪日龄与体重的实际对应数据，我们据此构建了猪的生长模型，通过对实际数据进行拟合，得到了猪的日龄-体重对应曲线。

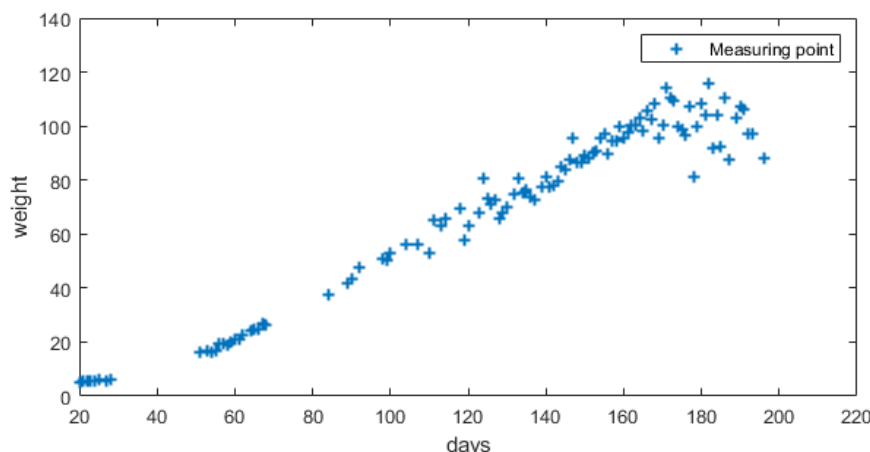


图 3 猪只的日龄-体重实际测量点分布图

图 3 为猪的部分实际测量数据图，采用的工具是 MATLAB 曲线拟合工具箱（Curve Fitting Tool），经过多重拟合方式和参数的筛选后，对这些数值点进行三次多项式的拟合，得到猪的日龄-体重对应曲线为图 4：

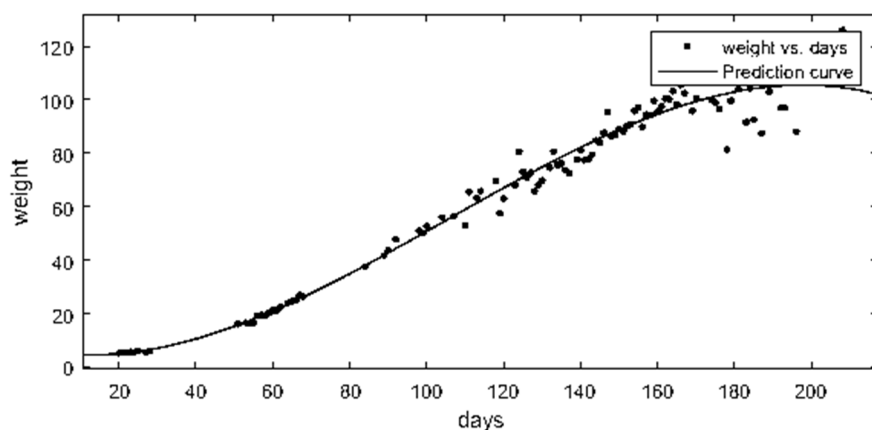


图 4 猪只的日龄-体重拟合曲线

根据实际经验和生物生长理论数据，猪的体重不能无限增加，而是在达到一定的体重后趋近于小幅波动，近似于不再变化。因此可以认为，在猪的日龄>200 天后，猪的体重将不再增加，且维持现有体重不变。因此，猪的生长模型的数学表达式构建如下：

$$W = \begin{cases} -0.00003185x^3 + 0.01006x^2 - 0.2398x + 6.039, & 0 \leq x \leq 200 \\ W(200), & x > 200 \end{cases}$$

其中, $W(200) = 105.679kg$ 是猪在第 200 天的预测体重。

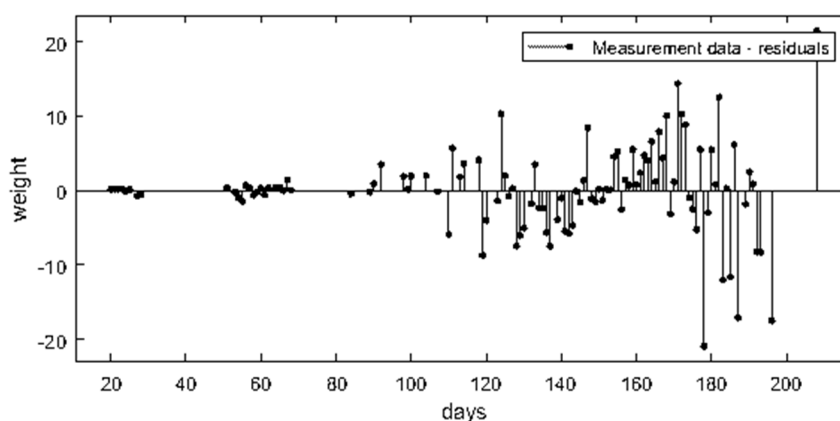


图 5 猪的日龄-体重拟合曲线残差图

上图 5 是实际值与预测点的误差, 所得曲线标准差为: 5.9413, 拟合具有一定的可信度。在本文后续解题过程中, 将基于上述曲线 W 对猪的体重进行预测。

4.3 配种与更新计划确定

图 1 已经具体分析了养猪场生产的流程, 下面将对其进行量化:

首先, 问题中给的条件是分娩率 r_1 为 88%, 哺乳期存活率 r_2 为 90%, 保育期存活率 r_3 为 97%, 育肥期存活率 r_4 为 98%, 仔猪公母比为 1:1, A/C/E 系公猪选种率 r_5 为 40%, A/B/C/D/F 系母猪的选种/自留率 r_6 为 65%, 可以计算出每次配种一只母猪能够得到多少只种猪。如: 1 只 A 母猪纯系交配可得到种母猪数量为:

$$n_{A,母} = \frac{1}{2} N_A r_1 r_2 r_3 r_4 r_6 = 2.84 \text{ 头}$$

式中, N_A 为 A 母猪产活仔数, 为 11-12 头, 本文设定为 11.5 头。类似地, 可以算出所有交配方式得到的种猪数量, 如下表 1

表 1 母猪与种猪产品的量化关系表

配种方式 (公*母)	种猪类型	数量 (头)
A*A	A 系公猪	1.73
A*A	A 系母猪	2.84
A*B	F 系母猪	2.84
B*B	B 系母猪	2.84
C*C	C 系公猪	1.39
C*C	C 系母猪	2.26
C*D	E 系公猪	1.39
D*D	D 系母猪	2.26

一个完整的生产种猪的过程，即从母猪配种到育肥猪选种，具有比较漫长的周期，一般在 8 个月以上。A/C/E 系公猪在 23 周龄选种，另外花费 2-3 周进行公猪预定，A/B/C/D 系母猪在 21 周进行选种或者自留测定，而 F 系母猪在 15 周进行测定。在制订配种计划时，每系种猪其母代配种一直到该种猪选完种所花费的时间总结如下表 2 所示：

表 2 各种系母猪生产周期表

种类	选种周龄/周	花费时间/天
A/C/E 系公猪	23+2.5	305
A/B/C/D 系母猪	21	273
F 系母猪	15	231

因此，在考虑销售计划时，仅能从一年中最后的两三个月的销售计划来倒推出之前两三个月时的需要配种母猪数量以及配种方式，而之后的配种计划则由明年（2020 年）种猪的销售计划决定。结合问题中提出的制定连续生产计划的要求，本模型假定 2020 年的种猪销售计划与 2019 年相同，这样就可以实现连续生产。

根据销售计划、种猪生产周期以及母猪与种猪产品共同决定下的量化关系，可以得到一年中每周对于各类母猪的需求，如下表 3 所示：

表 3 各类母猪的每周需求表（节选）

周次	A 公 A 母	B 公 B 母	A 公 B 母	C 公 C 母	D 公 D 母	C 公 D 母
1	0	6	20	0	0	6
2	0	6	16	0	0	6
3	0	7	6	0	0	6
4	0	12	6	0	9	6
5	0	9	6	0	7	11
6	0	8	6	0	7	12
7	0	9	5	0	7	12
8	0	9	5	0	7	12
9	0	9	5	2	7	10
10	0	8	5	5	6	9
11	0	7	7	5	5	9
12	0	6	8	5	5	9
13	0	6	9	5	4	9
14	2	0	25	0	0	15
15	2	0	25	0	0	15
16	2	0	25	0	0	15
17	2	0	25	0	0	15
18	0	0	20	3	6	13
19	0	0	12	3	13	14
20	0	0	12	3	13	14
...	...	...	...	...	...	...

养猪场生产种猪的理想状态就是按需生产，做到精确符合销售预测计划，但实际上存在着各种约束。例如产房约束，产房内正常母猪存栏总数不超过 $24 \times 6 = 144$ 头，而每头母猪须在产房里待 24 天，因此每天配种的母猪数量不能超过 $144/24 = 6$ 头，即每周配种的猪的数量不能超过 42 头。上表 3 中可以看出，有 13 周的配种数量大于了产房容纳量，而前 18 周的配种数量基本上不超过 25，这由种猪市场需求的季节不平衡性决定。为了尽量满足销售计划，本模型将每周配种数量超过产房容纳量的部分匀至配种任务比较少的周内，尽管其代价是提高了待售种猪的饲料成本。

显然，如果一直不将日龄较大的猪售卖出去，那么可能会使该种猪之后因为年龄过大而不符合售卖要求，故每个月种猪的销售将会选择满足条件的待售种猪中日龄较大的个体。因此虽然在计划上，某一周 n 配种数量可能分给之前并且隔其比较远的一周 $n - m$ ，但实际上是 i 周的配种任务给了 $n - 1$ 周，而 $n - 1$ 周的配种任务则移交给 $n - 2$ 周，这两者是等价的，所以不会出现待售种猪日龄过大的情况。

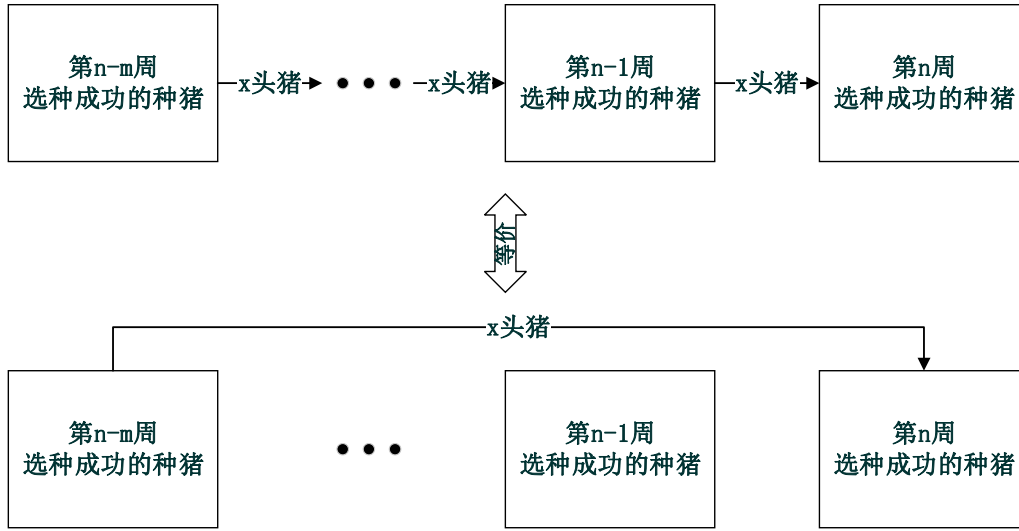


图 6 两种等价的调度模式

因此，这部分优化的目标函数是：

$$\min \sum_{j>i} u_{ij}(j-i),$$

$$\begin{cases} \sum_{k=1}^j x_{ik} \geq M_{ij}, j=1,2,\dots,52, i \text{ 为 } A,B,C,D; \\ \sum_{k \text{ 为 } A,B,C,D} x_{kj} \leq 42, j=1,2,\dots,52. \end{cases}$$

式中， u_{ij} 为第 j 周向第 i 周借的种猪的数量， x_{ij} 为第 j 周 i 类母猪的配种数量， M_{ij} 为第 j 周 i 类母猪的需求量。

由简单分析可知，采取就近原则，合理分配配种计划，使其满足产房约束，得到一个不包含生产自留母猪的配种计划：

表 4 调整后的配种计划表（节选）

周数	A	B	C	D	调整前总数	调整后总数
...	..	...	...	...	...	...
13	0	9	5	9	22	23
14	2	25	0	15	25	42
15	2	25	0	15	25	42
16	2	25	0	15	25	42
17	2	25	0	15	25	42
18	0	20	3	19	30	42
19	0	12	3	27	38	42
20	0	12	3	27	41	42
21	0	12	3	27	42	42
22	5	15	1	21	42	42
23	9	20	0	13	42	42
24	9	20	0	13	47	42
25	9	20	0	13	53	42
26	0	21	1	13	52	42
...	...	...	...	...	...	...
46	0	21	0	12	29	33
47	0	27	0	15	29	42
48	2	25	0	15	35	42
49	4	25	0	13	48	42
50	4	25	0	13	48	42
51	4	25	0	13	48	42
52	4	25	0	13	48	42

上述计划只是满足预测销售计划以及产房容纳约束。假设每一种母猪的数量都足够时，根据拟定的配种计划，可以得到每一周处于怀孕期的各种类母猪的数量，如下表 5 所示：

表 5 处于怀孕期的各种类母猪的数量表（节选）

周数	28	29	30	31	32	33	34	35	36	37	38
A	40	40	40	42	45	48	51	52	54	56	58
B	388	400	412	419	421	423	424	418	419	419	414
C	42	45	46	41	36	31	26	26	26	26	26
D	351	345	341	346	355	364	374	379	376	373	370

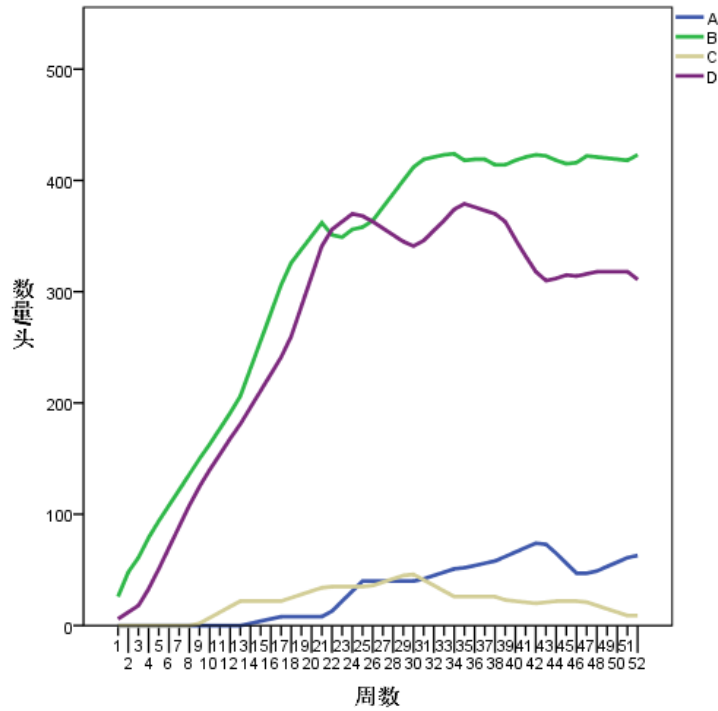


图 7 每周处于怀孕期的各种类母猪的数量图

由图 7 可知，A、B、C、D 系母猪的最大需求量分别为 74、424、46、379 头，而初始猪场中，生产母猪和后备母猪的各种类数量如下表：

表 6 各类母猪生产母猪与后备母猪数量表

类别	生产母猪数量	后备母猪数量
A	201	7
B	624	24
C	24	7
D	271	16

即使算上生产母猪以及后备母猪的总数量，C 系母猪与 D 系母猪都无法达到所需求的母猪数量要求。在本模型中，第一约束条件即为预测销售计划，制定的其他计划是服务于预测销售计划的。根据上述得到的对各类母猪数量的要求，本模型首先制定了一个各类母猪的增减计划：

表 7 增减计划表

A	-121
B	-34
C	+26
D	+109

虽然对于 B 系母猪的需求量只有 424 头，但是还需要保证生产母猪总数量大于 1100 头，因此只减少 B 系生产母猪 34 头。最后得到的生产母猪群体个种类的数量如下表：

表 8 平衡状态下生产母猪各种类数量

类别	生产母猪数量
A	80
B	590
C	50
D	380

根据题意,生产母猪更新的年更新率要达到 65%,除了不同类别母猪的更换,还有同类母猪内的更新,生产母猪的产生途径包括两种,一是自留母猪,二是购买。显然,为了最大化利润,应当尽量从第一个途径中得到生产母猪。

得到自留母猪的周期比较长,从母猪发情配种,到其生产的母猪自留,达到可配种的年龄,总共花费 $18+31=49$ 周,也就是说 2019 年第一周生产母猪完成配种,一直到第 50 周自留的母猪才能够成为后备母猪更新生产母猪,因此,实际上 2019 年用于生产自留母猪而确定的配种计划绝大多数是服务于 2020 年的生产母猪更新计划。

先忽略初始猪群的状态,考虑该养猪场的一个平衡的长久的状态,上表已经给出了生产母猪中各种类的数量,由年更新率 65%可以确定每年各类母猪的更新数量:

表 9 各类母猪更新数量

	A	B	C	D
更新数量(头)	52	384	33	247

分析可知,只需要考虑 B 类母猪与 D 系母猪的更新。因为 A 系与 C 系猪的纯系交配得到的仔猪用于销售的只是公猪,拥有足够数量的母猪可以用于自留,并且 A 系母猪和 C 系母猪对于更新母猪的需求不大,所以能够实现自给自足。

B 系母猪和 D 系母猪除了用于自留,很重要的用途就是用于销售。前面制定的配种计划也仅仅是满足销售目的,需要加入新的配种子计划。一年中比较空闲的时间是: 1-13 周、38-46 周,将 B 与 D 系纯系交配的任务分配到这些周,使每一周的总的配种数量大致相同。得到以下分配:

表 10 4-13 周加上自留配种后的计划表

周数	4	5	6	7	8	9	10	11	12	13
A*A	0	0	0	0	0	0	0	0	0	0
B*B	12	9	8	9	9	9	8	7	6	6
A*B	6	6	6	5	5	5	5	7	8	9
C*C	0	0	0	0	0	2	5	5	5	5
D*D	9	7	7	7	7	7	6	5	5	4
C*D	6	11	12	12	12	10	9	9	9	9
总数	33	33	33	33	33	33	33	33	33	33

表 11 40-46 周加上自留配种后的计划表

周数	40	41	42	43	44	45	46
A*A	4	4	4	4	1	0	0
B*B	4	4	6	6	8	9	1
A*B	12	11	8	8	8	8	20
C*C	2	2	2	2	1	0	0
D*D	3	4	5	5	6	7	0
C*D	8	8	8	8	9	9	12
总数	33	33	33	33	33	33	33

以上考虑的是一个平衡连续状态下的更新计划，但实际上，存在一端从初始状态慢慢变成连续平衡状态的过程，即 1-49 周，这段时间是没有自留母猪的产生的。因此只能从初始猪群里寻找销售计划之外的符合成为后备母猪条件的母猪。实际上，本模型之前确定的配种计划也是基于 2019 年 8 月份以后的销售计划。2019 年前 3/4 的销售计划是由初始猪只存栏满足的。

4.4 初始猪只存栏分析

从上述模型可知，2019 年 1-10 月 A、C、E 系种公猪，1-9 月 B、D 系种母猪以及 1-8 月半 F 系母猪的销售是由初始猪只存栏满足的，因此需要对初始猪只存栏进行分析。结果如下图：

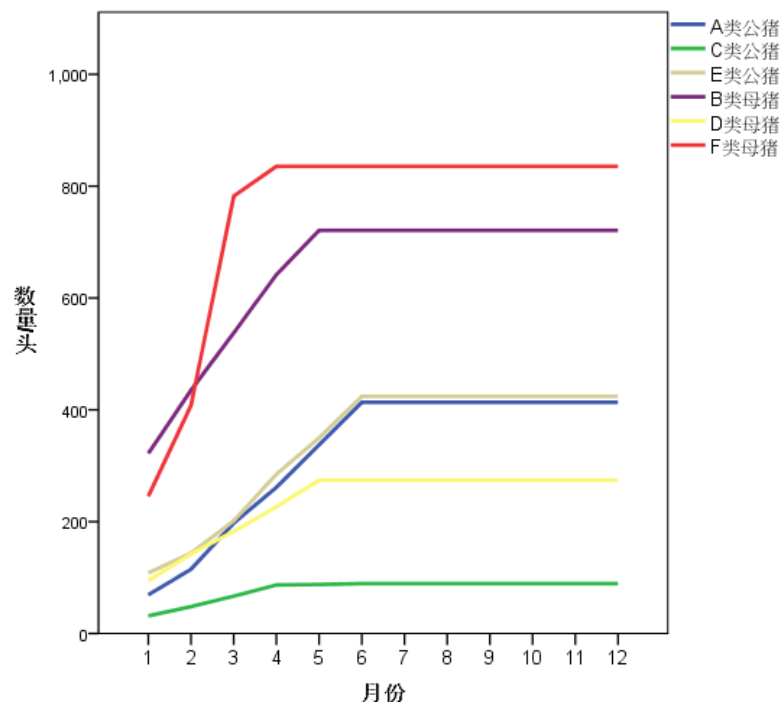


图 8 初始猪群种猪累计图

减去各系种猪每月销售量后，得到的余量表如下：

表 12 减去销售量后的每个月累计余量表

月份	1	2	3	4	5	6
A 系公猪	322	435	438	542	621	521
C 系公猪	95	22	62	107	54	54
E 系公猪	95	-2	42	-255	-405	-505
B 系母猪	69	98	180	176	252	308
D 系母猪	4	20	19	39	20	21
F 系母猪	55	6	-19	-12	-27	-29
月份	7	8	9	10	11	12
A 系公猪	521	521	421	421		
C 系公猪	54	54	54	54		
E 系公猪	-755	-1005	-1080	-1137		
B 系母猪	277	247	247			
D 系母猪	21	6	6			
F 系母猪	-105	-155				

由表可以看出 E 系公猪和 F 系母猪无法满足销售计划，且相差较大，但由于条件限制无法实时获得足够数量的种猪数量以供销售。

B 和 D 系母猪的余量比较大，因此可以作为自留母猪。其中 B 系母猪的余量为 421 头，D 系母猪的余量为 54 头。

除了上述两方面，还需要对 A 系母猪和 C 系母猪自留进行分析，这两类不需要参与销售，只需考虑是否有足够自留的数量，每个月可以参与配种的两类母猪的数量如下：（周表）

表 13 每周可以作为后备母猪的累计数量（节选）

周数	1	2	3	4	5	6	7	8	9	10	11	12	13
A 母：可配种	1	1	1	1	1	15	27	40	68	83	102.5	120.7	140.2
C 母：可配种	0	0	0	0	0	1	3	16	17	20	27.15	31.7	40.8

明显，有足够数量的 A/C 系猪用于自留。

4.5 确定更新计划

上文已经确定了各类生产母猪的增减计划，一共是交换了 135 头猪，还需要更新的猪的数量为 $1100 \times 65\% - 135 = 580$ 头，明显第一年 D 系生产母猪的数量是不够的，因此本文只考虑更新 A/B/C 系母猪，按比例更新，三类猪的更新数量为 64/475/40 只，另外还需考虑增加的生产母猪，各类生产母猪更新渠道如下：

表 14 更新生产母猪的渠道表

途径	初始后备母猪	初始猪只存栏	自留母猪	购买
A 系母猪	7	57	/	/
B 系母猪	24	421	30	/
C 系母猪	7	59	/	39
D 系母猪	16	54	/	/

其中 D 系生产母猪因为在第 20 周的需求量已经超过了现有生产母猪+后备母猪+初始猪只存栏里销售之外满足条件母猪的总和,而自留母猪需要等到 50 周之后, 因此只能选择在 20 周之前购买 39 头生产母猪。

还有一个要求是尽量快速清空后备母猪, 一是因为后备母猪存在最大容量 110 头, 二是能迅速解决老的生产母猪减少饲料成本。

生产公猪的更新考虑的就比较简单, 由于题中所给条件中没有提到种公猪可以用来自留, 因此本模型只考虑一个生产公猪更新渠道, 即购买。由于只有购买种公猪来更新生产公猪, 因此不存在上述对于生产母猪那样的考虑, 本文确定在第 45 周更新所有的生产公猪, 共 30 头, 不改变其群体构成。根据上述分析, 本模型得到的总的生产猪更新计划见附录 8。

4.6 确定配种计划

在各类生产母猪的更新渠道中, B 系生产母猪还需要 54 头母猪, 而 B 系母猪的数量是盈余的, 所以确定这 54 头猪由 1-3 周的配种自留产生。所以前三周的自留配种计划由此产生, 如下表 15 所示:

表 15 前三周加上自留配种后的计划表

周数	1	2	3
A*A	0	0	0
B*B	6	6	7
A*B	20	16	6
C*C	0	0	0
D*D	0	0	0
C*D	6	6	6
总数	32	28	19

而后面 4-52 周与第二年的前三周形成一个连续平衡的自留配种计划, 结合之前的分析得到的总的配种计划见附件 8。

4.7 计算机程序仿真

为了验证理论计算产生的配种和更新计划的可行性, 本文高度还原题目中的限制条件及运行机理, 以检查配种和更新事件是否会出错。

因为仿真的时间粒度是自然日, 程序中具有将周计划转换成天计划的模块化函数, 可以为配种及更新计划提供进一步的支撑。

除此以外, 根据配种计划表, 通过计算机仿真, 得到全年没有配种安排的天数共有 26 天。

五、基于遗传算法的副产品销售方案确定模型

5.1 建立初始矩阵

上述模型得到了一年的配种计划以及生产猪更新计划，结合 2019 年预测销售计划以及初始猪只存栏，先不考虑副产品的售卖，可以得到每一时刻猪场的状态，包括各类猪的数量、每只猪的状态等。

本文建立种猪实体与副产品实体的转换渠道，其中仅生产猪更新时由种猪转变成副产品。种猪实体和副产品实体分别包括以下属性：

表 16 两类实体的属性表（左副产品，右种猪）

属性名	含义	属性名	含义
<i>Number</i>	编号	<i>Number</i>	编号
<i>Ini_age</i>	初始日龄	<i>Ini_age</i>	初始日龄
<i>state</i>	状态	<i>state</i>	状态
<i>age</i>	当前日龄	<i>age</i>	当前日龄
<i>weight</i>	题中	<i>weight</i>	题中
		<i>Kind</i>	种类

其中状态包含 6 种状态，分别是：已卖、还未出生、20-25kg 不能卖、可卖、还没有被更新、死亡。

种猪与副产品的来源以及影响因素如下：

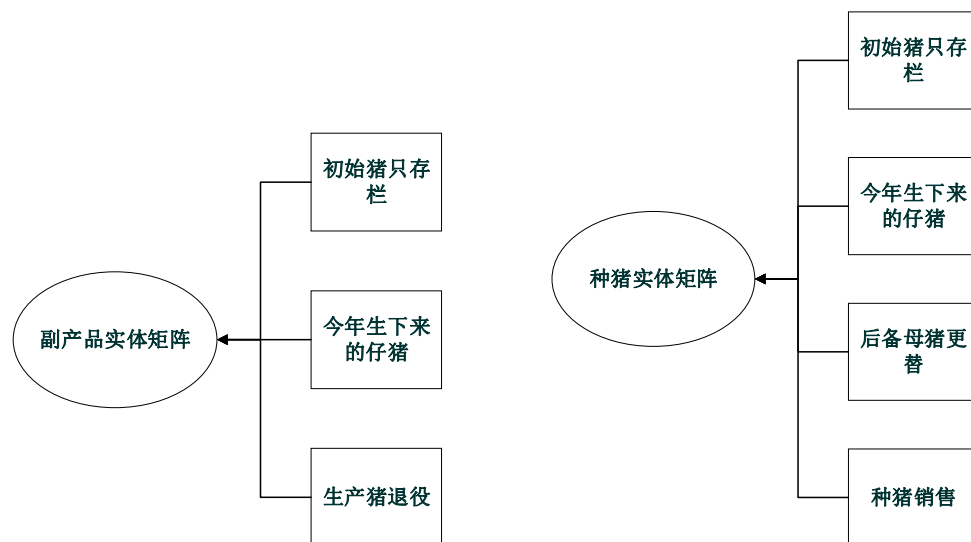


图 9 种猪与副产品的来源以及影响因素

根据问题中给的初始猪只存栏，种猪选种率以及自留测定率，首先确定每只母猪的实体属性，判断流程如下：

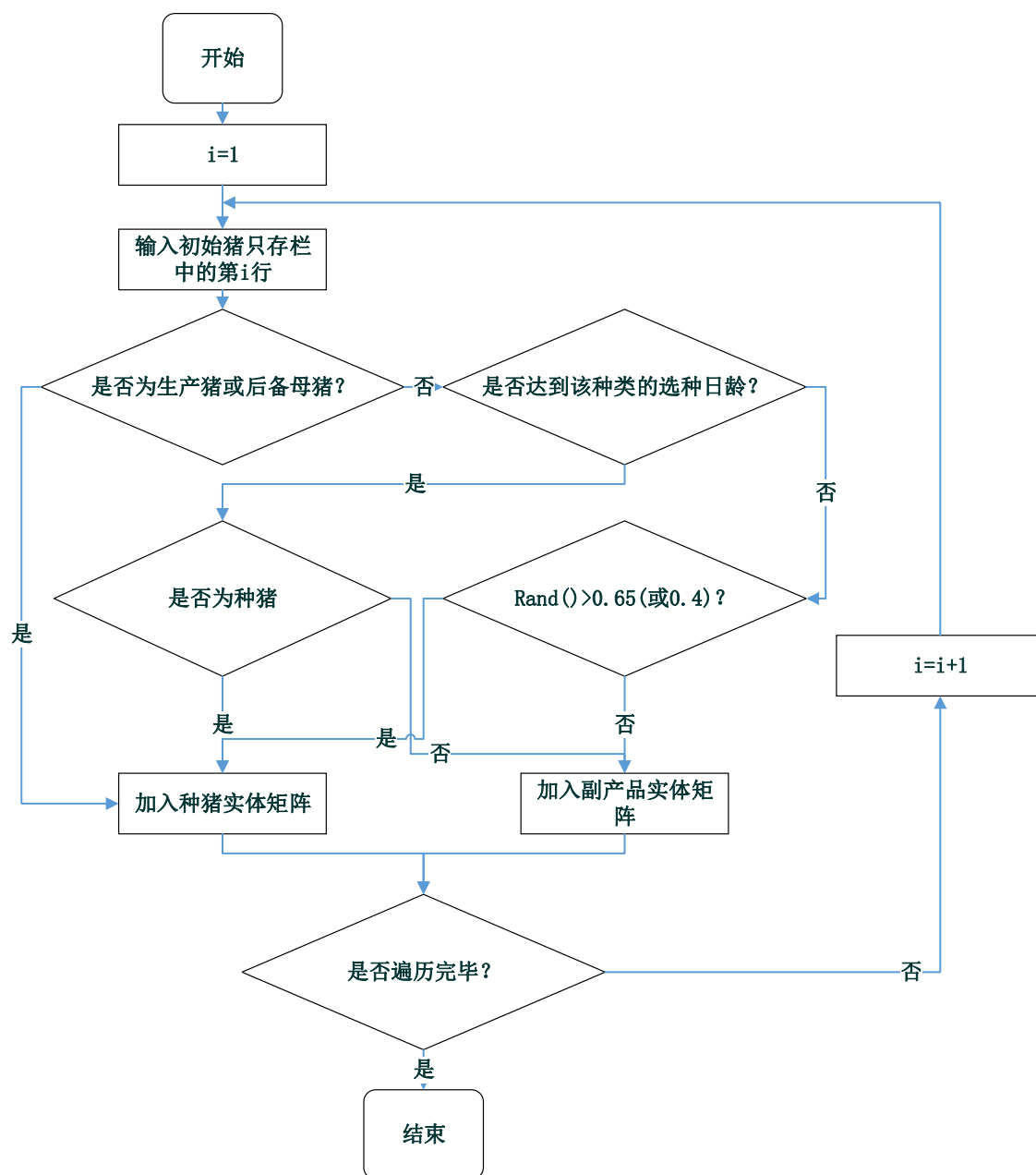


图 10 实体判断流程图

考虑完初始猪只存栏里的所有猪后，再确定由今年生产的猪的状态和实体属性。对于配种计划中的每一只母猪，其经过一个生产周期后得到的种猪和副产品的数量是确定的，具体如下图：

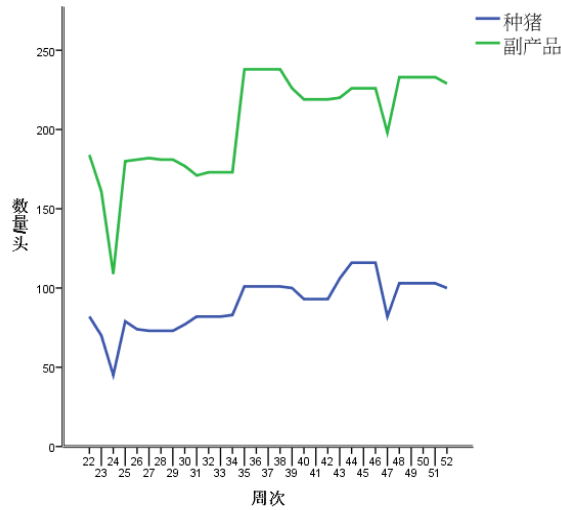


图 11 2019 年出生种猪和副产品数量图

因此能够在仔猪出生的时候确定多少新产生的种猪以及副产品的数量，同时还可以知道它们的出生日期、状态以及种类。本模型在时间 $t=0$ 设置没有出生的猪的初始日龄设置为负数，值为出生日期减去初始日期。

除了初始猪只和未出生仔猪外，副产品的第三个来源为生产猪更新退下来的老生产猪。它们的日龄足够大，由生长模型可知，猪在到达一定的日龄后体重保持平稳，因此可以忽略日龄因素，直接设置为 1000 即可。根据更新计划，当到达确定的更新日期时，将该猪进行实体转换，变为副产品。

生产猪更新还会影响种猪矩阵，具体过程是后备母猪代替生产母猪，会使种猪数量减少，但如果是从外面买的新生产猪，则不会影响种猪矩阵。除了生产猪更新，还有一个因素会使种猪数量减少，那就是每个月种猪的销售，本文假定月销售量按周平均分，得到周销售量，得到每周的种猪减少量。

由上述步骤可以得到种猪和副产品两个初始矩阵。

5.2 目标函数

本问题提出来的总目标函数实现利润的最大化，在第 4 部分的问题抽象中也说明了猪场利润的具体涵义。而本部分的模型是建立在已经确定了猪场配种计划和更新计划的基础上的，配种计划和更新计划就已经确定了主产品种猪销售的利润，所以在这一部分只考虑副产品的利润，具体数学形式如下：

$$L_{\text{副}} = R_{\text{副}} - C_{\text{运输}} - C_{\text{副产品饲养}}$$

具体的算法流程图如下：

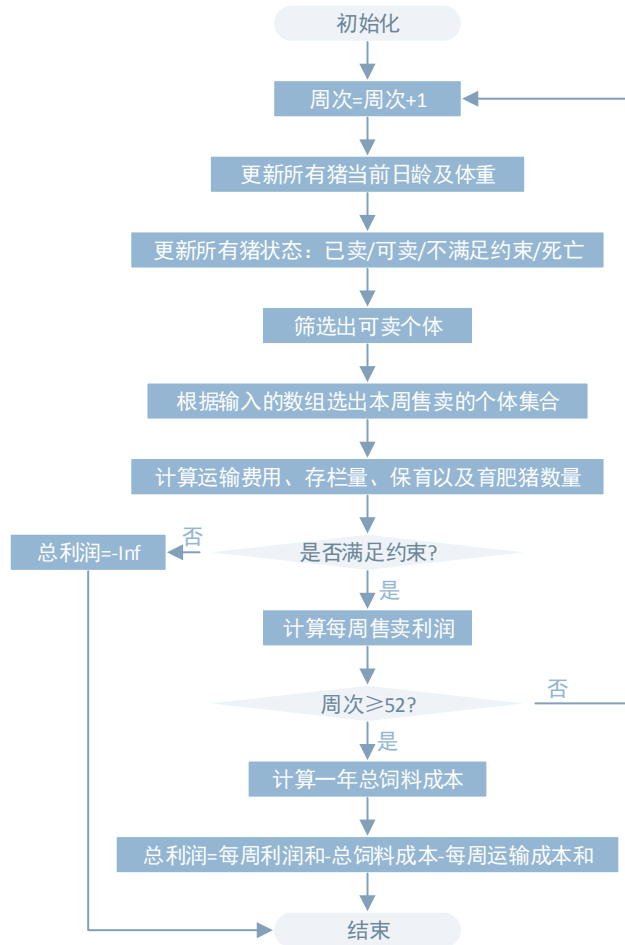


图 12 计算适应值流程图

5.3 解的形式

需要求解的变量是一年的副产品销售计划（以周为单位）。其形式如下：

$$Y = \{Y^1, Y^2\},$$

$$Y^1 = (y_1^1, y_2^1, \dots, y_n^1)^T,$$

$$Y^2 = (y_1^2, y_2^2, \dots, y_{52}^2)^T.$$

其中， Y^1 是一个 $1 \times n$ 的数组，表示所有副产品猪的售卖顺序，即，如果 y_1^1 和 y_2^1 同时满足售卖条件，id 号为 y_1^1 的猪将先于 id 号为 y_2^1 的猪被售卖。 Y^2 是一个 1×52 的数组，表示每周售卖的猪的数量。 $y_1^2 = 100$ 表示在第一个月将有 100 只副产品猪被售卖。

由所得到的解以及初始矩阵，即可算出副产品的利润，即该解的目标函数值。特别注意解包含的两个向量只能在内部进行遗传算法的算子操作，不能混合在一起。

5.4 约束条件

除了求解模型一过程中已经满足了的约束，剩下的约束条件都要在本模型求解过程中得到满足，包括如下几项

- 猪总数量约束：种猪以及副产品在每个时间节点的数量和大于等于 9000 头；
- 猪舍容纳约束：任何时刻保育猪以及育肥猪的数量不超过限制；
- 运输能力约束：本模型是以周为时间计算单位的，一共有两辆车，每辆车一周最多运行三次，所以每周的副产品销售量不大于两辆车一周的最大运输量之和。

5.5 建模过程

本模型运用的是遗传算法来对副产品销售计划进行优化，具体过程如下：

Step1	确定遗传算法的各类参数：种群个数，交叉概率，变异概率，选择算子，交叉算子，变异算子； 两个种群有不同的选择交叉变异概率；
Step2	种群初始化，随机生成 40 个初始解；
Step3	开始迭代，进行选择、交叉操作；
Step4	进行变异操作；
Step5	将每个解即副产品销售计划结合初始矩阵进行时间仿真，可以得到一个具体的仿真过程以及每个时间节点猪场的状态，判断是否满足约束条件，若不满足，目标函数值设为-1；若满足约束条件，计算目标函数值，目标函数值作为该个体的适应值；
Step6	计算该次迭代中适应值最大的解，与之前记录的最大适应值进行对比，对最大适应值和最优解进行更新；
Step7	迭代数+1，判断是否到达最大迭代数，若到达，跳出循环，输出最优解和最大适应值；若没有达到，返回第 3 步； 交叉算子：随机选出数组中的两个长度相同的片段，交换两个片段的位置； 变异算子：对数组内部的随机片段重新进行排序。

具体流程如下图：

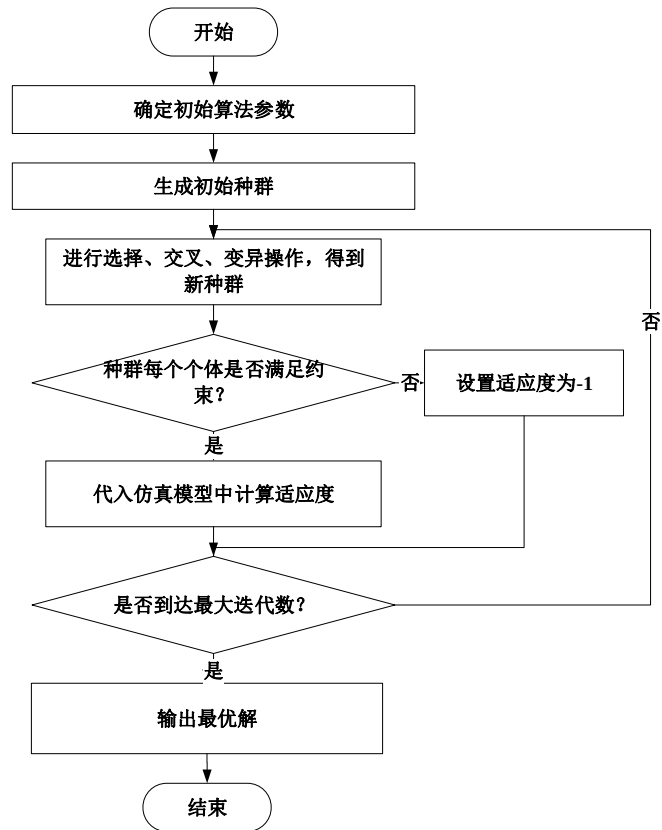


图 13 遗传算法流程图

5.6 副产品销售计划结果展示

由于时间限制，我们只迭代了 30 次来寻找最优解。30 次迭代得到的最大收入变化曲线如图 14 所示。

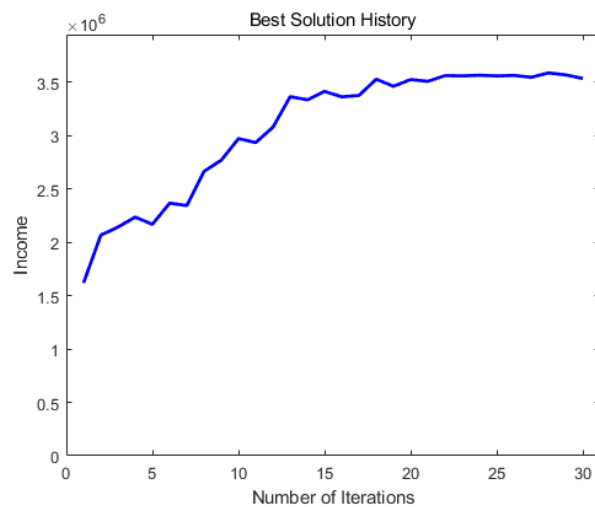


图 14 遗传算法最优解变化图

由于计算适应值函数的过程中，副产品猪的随机死亡可能会导致适应值的

些微波动，因此导致了最大适应值函数变化曲线的波动。总体上看，30代时，最大适应值已趋于平缓，其具体值为 3.5854×10^6 。并得出了最优解 Y_{best}^1 和 Y_{best}^2 。将最优解代入适应值求解函数中，求得各类副产品每周的销售计划如图15所示，详见附件9。

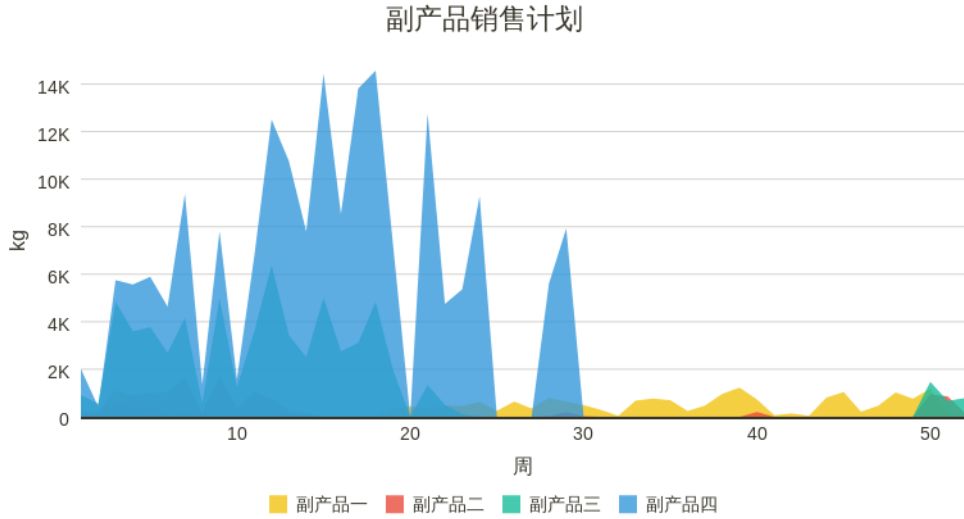


图 15 每周副产品销售计划

5.7 饲料消耗量求解与结果展示

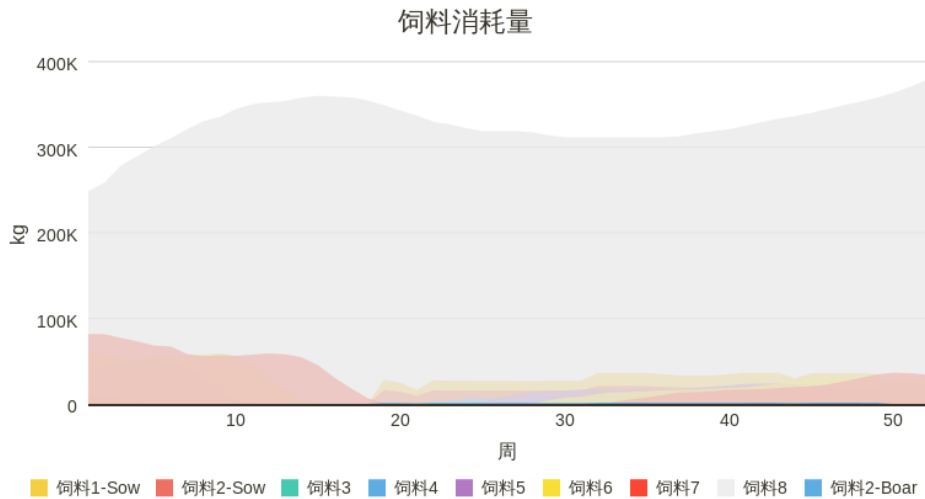


图 16 每周各类饲料消耗量

结合前一问的模型能够得到代入最优副产品销售计划的猪场仿真数据，即可以知道每一只猪在某个时间点的具体状态，再通过饲料消耗量与日龄关系函数即可计算每周各类饲料消耗量。

在前几问的求解过程中，可以得知一年中所有种猪产品和副产品的出生时间以及售卖时间，据此可通过饲喂程序获得每头猪每日的各类饲料消耗量，从而计算出每周各类饲料消耗量，详见附件10。

六、模型优缺点

优点：

1. 本模型将养猪场连续生产计划制定的实际问题进行了抽象与建模求解，在解决这类型的实际问题方面具有重要的指导意义；
2. 求解过程严谨，逻辑性强，可信度高；
3. 探索了一种解决复杂优化问题的新方向，即将解进行拆分，考虑两个优化问题的求解；
4. 模型在计算过程中对下一年的售卖计划进行了考虑，而不是仅考虑本年销售与收益，体现了该模型的连续性，使模型更加贴近实际情况；
5. 理论模型与计算机仿真相辅相成，反向推理和正向运算，验证解的可靠性，体现探究的严谨以及模型的完备性。

缺点：

1. 在本模型第一部分得到的解没有进行充分验证，无法确定是否是全局最优还是局部最优；
2. 为了简化求解过程，理想化了一些设定，如本模型中的生长模型仅仅是基于部分测量数据的简单拟合结果，而在现实中，不同种类的猪的生长曲线可能不同，可以对不同类型，不同情况下的猪的生长模型进行探讨；
3. 已有研究和文献基础薄弱，模型搭建比较简单，可能存在更好的算法求解问题；
4. 本模型中假设已确定 2019 年种猪销售计划，根据对本问题的分析可得知实际情况不能满足销售计划需求，可以考虑在对副产品销售计划进行优化的同时加入对种猪销售计划的优化；
5. 题目设定下的产房、保育舍和育肥舍数量都已固定，考虑到在实际中养猪场要增加收益，扩大养猪场的规模也是一个有效且效果客观的选择。因此可以对产房、保育舍和育肥舍的存栏量适量增减问题进行研究，已辅助养猪场获得更大利润。

七、问题的进一步拓展

1. 本模型中的生长模型仅仅是基于部分测量数据的简单拟合结果，并且而在现实中，不同种类的猪的生长曲线可能不同，可以对不同类型，不同情况下的猪的生长模型进行探讨；
2. 本模型中假设已确定 2019 年种猪销售计划，根据对本问题的分析可得知实际情况不能满足销售计划需求，可以考虑在对副产品销售计划进行优化的同时加入对种猪销售计划的优化；
3. 本题中的产房、保育舍和育肥舍数量都已固定，考虑到在实际中养猪场要增加收益，扩大养猪场的规模也是一个有效且效果客观的选择。因此可以对产房、保育舍和育肥舍的存栏量适量增减问题进行研究，已辅助养猪场获得更大利润；
4. 通过广泛地阅读文献，我们发现题意中提及的**连续生产管理**模式亟待改

进为**批量批次生产模式**，即每周集中的有规律生产模式。该模式已经广泛应用于西欧国家，实现工厂化大规模生产，其能有效地阻断疾病传播、优化管理、提高人员效率、周期性空栏时间有利于卫生消毒工作的开展等等。对比本文中连续生产的配种环境每年仅有 26 天空闲期，该模式可以极大的解放劳动力。如果针对这一模式更进一步研究，批次生产的生产销售计划可能对于我国农业养殖业的开拓性探索具有更大的建设性意义。

八、参考文献

- [1] 李桦. 生猪饲养规模及其成本效益分析[D]. 西北农林科技大学, 2007.
- [2] 徐如海, 胡锦涛, 翁经强,等. 连续日称重杜洛克公猪的生长曲线分析[J]. 浙江农业学报, 2007, 19(5):343-345
- [3] 宋美娇. 山东省生猪饲养成本效益研究[D]. 山东农业大学, 2016.
- [4] 吴孟达, 成礼智, 吴翊, 毛紫阳, 王丹. 数学建模教程[M]. 国防科学技术大学出版社, 2013.
- [5] 郁磊. MATLAB 智能算法 30 个案例分析[M]. 北京航空航天大学出版社, 2011.
- [6] 谢秉磊, 孙毅, 李荣喜. 求解配送\收集旅行商问题的遗传算法[J]. 陕西理工大学学报(自然科学版), 2002, 18(1):70-74.
- [7] 熊翠, 吴慧萍, 李波. 多旅行商问题遗传算法求解及其改进[C]// 中国智能计算大会. 2010.
- [8] 叶多福, 刘刚, 何兵. 一种多染色体遗传算法解决多旅行商问题[J]. 系统仿真学报, 2019(1).
- [9] 蔡良伟, 李霞, 张基宏. 用带蚁群搜索的多种群遗传算法求解作业车间调度问题[J]. 信息与控制, 2005, 34(5):553-556.

九、附录

9.1 附录一：支撑材料说明

文件名	备注
附件 8.xlsx	每周配种计划，及生产公猪、生产母猪的更新计划
附件 9.xlsx	副产品的每周销售计划
附件 10.xlsx	饲料消耗量
/仿真程序/report.html	附件 6-猪只存栏的探索性分析
/仿真程序/simulation.py	仿真主程序（可直接运行，测试运行环境为 Python 3.7.1 + Windows7(OS)）
/仿真程序/result.txt	记录配种和更新信息的日志文件
/MATLAB 程序/ cal_food.m	计算每天饲料成本
/MATLAB 程序/ mean1.m	计算日龄-体重拟合曲线
/MATLAB 程序/ mstsp_ga.m	遗传算法求解最大适应值
/MATLAB 程序/ second.m	根据输入计算适应值（利润）
/MATLAB 程序/ second_last.m	得到最优解之后计算最终作为副产品售卖的猪只表
/MATLAB 程序/ zhongzhunum.m	计算最终作为种猪售卖的猪只表
/MATLAB 程序/ last_fuchanpin	最终作为副产品售卖的猪只表
/MATLAB 程序/ last_zhongzhu	最终作为种猪售卖的猪只表
/MATLAB 程序/ matlab.mat	遗传算法最优解参数
/MATLAB 程序/ fujian9.mat	副产品销售计划
/MATLAB 程序/ result_food.mat	每日/每周饲料成本统计表

9.2 附录二：MATLAB 程序：遗传算法求解最大值

```
#mtsp_ga.m
%遗传算法求解最大适应值
clear;
clc
close all
load('fuchanpin.mat');

for k =0:7
    switch k
        case 0
```

```

        xy = 10*rand(20,2);
    case 1
        N = size(xy,1);
        a = meshgrid(1:N);
    case 2
        salesmen = 53;
    case 3
        min_tour = 3;
    case 4
        pop_size = 40;
    case 5
        num_iter = 30;
    case 6
        show_prog = 1;
    case 7
        show_res = 1;
    otherwise
    end
end
% Verify Inputs
[N,dims] = size(xy);
n=11021;
% Sanity Checks
salesmen = max(1,min(n,round(real(salesmen(1)))));
min_tour = max(1,min(floor(n/salesmen),round(real(min_tour(1)))));
pop_size = max(8,8*ceil(pop_size(1)/8));
num_iter = max(1,round(real(num_iter(1))));
show_prog = logical(show_prog(1));
show_res = logical(show_res(1));
% Initializations for Route Break Point Selection 生成初始种群
num_brks = 52;
dof = n - min_tour*salesmen;          % degrees of freedom
addto = ones(1,dof+1);
for k = 2:num_brks
    addto = cumsum(addto);
end
cum_prob = cumsum(addto)/sum(addto);
for k = 1:pop_size
    pop_rte(k,:) = randperm(length(pigdata),length(pigdata));
    pop_brk(k,:) = round(220*rand(1,num_brks));
end
% Select the Colors for the Plotted Routes
clr = [1 0 0; 0 0 1; 0.67 0 1; 0 1 0; 1 0.5 0];
if salesmen > 5

```

```

        clr = hsv(salesmen);
    end
    % Run the GA
    global_min = -1;
    total_dist = zeros(1,pop_size);
    dist_history = zeros(1,num_iter);
    tmp_pop_rte = zeros(8,n);
    tmp_pop_brk = zeros(8,num_brks);
    new_pop_rte = zeros(pop_size,n);
    new_pop_brk = zeros(pop_size,num_brks);
    if show_prog
        pfig = figure('Name','MTSP_GA | Current Best Solution','Numbertitle','off');
    end
    for iter = 1:num_iter
        iter
        % Evaluate Members of the Population
        for p = 1:pop_size            %计算返回值
            pop_size;
            total_dist(p) = second(pop_rte(p,:),pop_brk(p,:));
        end
        % Find the Best Route in the Population
        [min_dist,index] = max(total_dist);
        dist_history(iter) = min_dist;
        if min_dist > global_min
            global_min = min_dist;
            opt_rte = pop_rte(index,:);
            opt_brk = pop_brk(index,:);
            rng = [[1 opt_brk+1];[opt_brk n]]';
        end
        % Genetic Algorithm Operators 遗传算法算子
        rand_grouping = randperm(pop_size);
        for p = 8:8:pop_size
            rtes = pop_rte(rand_grouping(p-7:p),:);
            brks = pop_brk(rand_grouping(p-7:p),:);
            dists = total_dist(rand_grouping(p-7:p));
            [ignore,idx] = max(dists);
            best_of_8_rte = rtes(idx,:);
            best_of_8_brk = brks(idx,:);
            rte_ins_pts = sort(ceil(n*rand(1,2)));
            I = rte_ins_pts(1);
            J = rte_ins_pts(2);
            for k = 1:8 % Generate New Solutions
                tmp_pop_rte(k,:) = best_of_8_rte;%最优染色体
                tmp_pop_brk(k,:) = best_of_8_brk;
            end
        end
    end
end

```

```

        switch k
            case 2 % 片段交换
                tmp_pop_rte(k,I:J) = fliplr(tmp_pop_rte(k,I:J));
            case 3 % 第一染色体变异
                %tmp_pop_rte(k,[I J]) = tmp_pop_rte(k,[J I]);
                tmp_pop_rte(k,:) =
randperm(length(pigdata),length(pigdata));
            case 4 % 第一染色体变异
                %tmp_pop_rte(k,I:J) = tmp_pop_rte(k,[I+1:J I]);
                tmp_pop_rte(k,:) =
randperm(length(pigdata),length(pigdata));
            case 5 % 第二染色体变异
                tmp_pop_brk(k,:) = round(220*rand(1,num_brks));
            case 6 % 第一染色体片段交换， 第二染色体变异
                tmp_pop_rte(k,I:J) = fliplr(tmp_pop_rte(k,I:J));
                tmp_pop_brk(k,:) = round(220*rand(1,num_brks));
            case 7 % 第一染色体变异， 第二染色体变异
                %tmp_pop_rte(k,[I J]) = tmp_pop_rte(k,[J I]);
                tmp_pop_rte(k,:) =
randperm(length(pigdata),length(pigdata));
                tmp_pop_brk(k,:) = round(220*rand(1,num_brks));
            case 8 % 第一染色体变异， 第二染色体变异
                %tmp_pop_rte(k,I:J) = tmp_pop_rte(k,[I+1:J I]);
                tmp_pop_rte(k,:) =
randperm(length(pigdata),length(pigdata));
                tmp_pop_brk(k,:) = round(220*rand(1,num_brks));
            otherwise % Do Nothing
        end
    end
    new_pop_rte(p-7:p,:) = tmp_pop_rte;
    new_pop_brk(p-7:p,:) = tmp_pop_brk;
end
pop_rte = new_pop_rte;
pop_brk = new_pop_brk;
end
if show_res
    plot(dist_history,'b','LineWidth',2);
    title('Best Solution History');
    set(gca,'XLim',[0 num_iter+1],'YLim',[0 1.1*max([1 dist_history])]);
end
% Return Outputs
varargout{1} = opt_rte;
varargout{2} = opt_brk;
varargout{3} = min_dist;

```

```
second(best_of_8_rte,best_of_8_brk);
min_dist;
```

9.3 附录三：MATLAB 程序：计算适应值

```
#second.m
```

```
%根据输入计算适应值（利润）
```

```
function sum_all = second(sellpop,sellnumber)
```

```
    load('daycost.mat');          %饲料价格表
```

```
    load('fuchanpin.mat');        %最终成为副产品的所有猪的表：% 第一列：
```

序号

% 第

二列：猪的初始日龄

% 第

三列：猪的状态：-1 已卖；-2 还未出生；0：20-25kg 不能卖；1 可卖；-3：还没有从生产母猪退役下来；-4 死亡

% 第

四列：当前日龄

% 第

五列：当前体重

```
    load('price.mat');           %售卖价格表
```

```
    load('num_zhongzhu.mat');    %种猪数量表：第一列：总数量，第二列：
```

保育猪数量，第三列：育肥猪数量

```
    transfee = [500,550,1050,1550,1600,2100,2600,2650,3150];
```

```
    daymonth(1:31) = 1;
```

```
    daymonth(32:59) = 2;
```

```
    daymonth(60:90) = 3;
```

```
    daymonth(91:120) = 4;
```

```
    daymonth(121:151) = 5;
```

```
    daymonth(152:181) = 6;
```

```
    daymonth(182:212) = 7;
```

```
    daymonth(213:243) = 8;
```

```
    daymonth(244:273) = 9;
```

```
    daymonth(274:304) = 10;
```

```
    daymonth(305:334) = 11;
```

```
    daymonth(335:365) = 12;
```

```
sold = [];
```

```
death = [];
```

```
sumfee=0;
```

```
sumbenefit = 0;
```

```
sum_all = 0;
```

```
% pigdata
```

```

%计算更新母猪的饲料成本
cost_beforeupdate = sum(daycost(updateday(:,3),2));
cost_beforestart = sum(daycost(updateday(:,2),2));
cost_betweenstartandupdate = cost_beforeupdate - cost_beforestart;

for day = 4:7:365
    week=(day+3)/7;
    pigdata(:,4) = pigdata(:,2)+day; %更新当前日龄
    pigdata(:,5) = (-3.185e-05*(13^3)+0.01006*(13^2)-0.2398*13+6.039).*(pigdata(:,4)<=13) + (-3.185e-05*(pigdata(:,4).^3)+0.01006*(pigdata(:,4).^2)-0.2398*pigdata(:,4)+6.039).*(pigdata(:,4)>13&pigdata(:,4)<=200) + (-3.185e-05*(200^3)+0.01006*(200^2)-0.2398*200+6.039).*(pigdata(:,4)>200); %更新当前体重

    %% 更新状态
    for i = 1:length(pigdata)
        id = pigdata(i,1);

        if (pigdata(id,3) ~= -1) & (pigdata(id,3) ~= -4) % 考虑死亡
            if pigdata(id,4) <= 24 & pigdata(id,4) >= 18 %分娩哺乳
                if rand()>0.9
                    pigdata(id,3) = -4;
                    death = [death;pigdata(id,:)];
                end
            elseif pigdata(id,4) <= 73 & pigdata(id,4) >= 67 %保育期
                if rand()>0.97
                    pigdata(id,3) = -4;
                    death = [death;pigdata(id,:)];
                end
            elseif pigdata(id,4) <= 164 & pigdata(id,4) >= 158 %育肥期
                if rand()>0.98
                    pigdata(id,3) = -4;
                    death = [death;pigdata(id,:)];
                end
            end
        end

        if (pigdata(id,3) ~= -1) & (pigdata(id,3) ~= -4) %先把没卖的/没死的都更新为“未出生”
            pigdata(id,3) = -2;
        end
    end
end

```

```

        end
        if (pigdata(id,3) == -2) & (pigdata(id,4)>0)           %将未卖的
猪根据日龄进行更新：是否出生
            pigdata(id,3) = 1;
        end
        if (pigdata(id,3) == 1) & (pigdata(id,5)>20) & (pigdata(id,5)<=25) %除
去 20-25kg 不能卖的
            pigdata(id,3) = 0;
        end
    end
    end
    [canchoose,~] = find(pigdata(:,3) == 1);           %选出可卖
的
    canchoose = pigdata(canchoose,:);

    %% 重量约束条件下选择卖的，并计算每周运输费用
    AAA = sellpop';
    for i = 1:length(AAA)
        if isempty(find (canchoose(:,1) == AAA(i)))
            AAA(i) = -1;
        end
    end
    AAA = AAA(AAA~-1);
    AAA = AAA(1:sellnumber(week));

    for i = 1:length(sellpop)
        if ~isempty(find (sellpop(i) == AAA))
            sellpop(1,i) = -1;
        end
    end
    sellpop = sellpop(sellpop~-1);

    bechosed=unique(AAA);
    choose= pigdata(bechosed,:);           %取卖
出去的
    weeksumweight = sum(choose(:,5));           %计
算本轮运输总重量
    if weeksumweight >40500
        sum_all = -1;
        break;
    end
    if weeksumweight>0
        fee = transfee(ceil(weeksumweight/9000*2));           %计算运
输费用
    else

```

```

        fee = 0;
    end
    sumfee = sumfee+fee;
    %% 统计更新

    temp = pigdata(pigdata(:,3)~= -1,:); %未卖未
    死的猪表
    temp = temp(temp(:,3)~= -4,:);
    number_unsoldundeath =
length(temp); %>9020
    if number_unsoldundeath + num_zhongzhu(week,1) < 9020
        sum_all = -1;
        break;
    end

    baoyu = temp(temp(:,4)>21,:); %保育
    猪表
    baoyu = baoyu(baoyu(:,4)<=70,:);
    number_baoyu =
length(baoyu); %<2675
    if number_baoyu + num_zhongzhu(week,2) >2675
        sum_all = -1;
        break;
    end

    yufei = temp(temp(:,4)>70,:); %育肥
    猪表
    yufei = yufei(yufei(:,4)<=161,:);
    number_yufei =
length(yufei); %<5657
    if number_yufei + num_zhongzhu(week,3) >5657
        sum_all = -1;
        break;
    end

    sold = [sold;choose]; %更新
    已卖副产品表
    clear choose;
    pigdata(bechosed,3)=-1; %总
    表中标记被卖出去的

    %% 计算每周利润
    month = daymonth(day);
    for i = 1:length(bechosed)

```

```

weight = pigdata(bechosed(i),5);
if weight<=20
    y = price(1,month)*weight;
elseif weight>20 & weight<=25
    y = 0;
elseif weight>25 & weight<=50
    y = price(2,month)*weight;
elseif weight>50 & weight<=100
    y = price(3,month)*weight;
elseif weight>100
    y = price(4,month)*weight;
end
sumbenefit = sumbenefit + y;
end

end
%% 计算总饲料成本
if sum_all ~= -1
    [unsold,~] = find(pigdata(:,3) ~= -1 & pigdata(:,3) ~= -4); %选出
    没有被卖掉的
    unsold = pigdata(unsold,:);
    %没有被卖掉的饲料总价
    cost_unsold = sum(daycost(unsold(:,4),2));
    cost_death = sum(daycost(death(:,4),2));
    cost_sold = sum(daycost(sold(:,4),2));
    cal_past = pigdata(pigdata(:,2)>0,:);
    cost_past = sum(daycost(cal_past(:,2),2));

    %更新的母猪：除去被淘汰下来之前的饲料量
    sumcost = cost_unsold + cost_sold - cost_past - cost_betweenstartandupdate;

    sum_all = sumbenefit-sumcost-sumfee;
end

```

