



湖南省高校第四届

研究生数学建模竞赛

学 校	国防科技大学
-----	--------

参赛队号	201918001012
------	--------------

	1. 杨家伟
--	--------

队员姓名	2. 孔令发
------	--------

	3. 李 凡
--	--------



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）：B 题

我们的参赛报名号为（如果组委会设置报名号的话）：201918001012

所属学校（请填写完整的全名）：国防科技大学

参赛队员（打印并签名）： 1. 杨家伟

2. 孙金发

3. 李凡

指导教师或指导教师组负责人(打印并签名)：

日期：2019 年 10 月 31 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： 猪场配种计划及副产品销售方案制定

摘 要：

猪场通过制定合理的生产、销售计划可以充分利用现有资源实现利润最大化。本文在综合考虑猪只生产规律、猪舍容量、运输能力等约束条件和配种、产仔、销售、更新等环节，通过建立三个模型，研究了猪场配种计划以及副产品销售方案制定等问题。

模型一共分三步实现了配种计划的制定。首先根据猪只生长规律及种猪销售计划制定状态更新函数，状态更新函数可给出每天所有猪的日龄及所处的状态。根据状态更新函数，可以判断出每天的待售种猪是否满足当天的销售计划，当不满足时需要通过反推确定反推时间点的配种方案。这样得到的配种方案只是考虑种猪销售计划的“初步配种方案”；其次在考虑生产公、母猪的更新的情况下，修正上一步得到的“初步配种方案”。更新生产猪时，由于生产公猪数量较少，可使用现存待售种猪更换；但需要更新的生产母猪数量较多，现存的后备母猪不足以支撑其更新，所以需要在“初步配种方案”的基础上继续增加配种以满足生产母猪的更新需求。经过这一步得到的配种计划我们将其称为“初步修正的配种计划”；最后考虑要为后续副产品的销售留出足够的裕度，保证猪场的持续盈利，所以要在满足猪舍最大存栏上限的前提下继续增加配种。模型通过合理的验算，确定保证每天均有三窝哺乳仔猪出生，进而通过时间反推可得到“最终配种计划”。经模型检验我们给定的配种计划可满足种猪销售计划、生产猪的更新计划并实现产能最大化，并且由于同时给出了 2019 年全年的配种计划，也为 2020 年种猪的销售留出了很大裕度。

模型二用来制定每周副产品的销售方案，以保证利润最大化。在模型一给定配种计划，并考虑副产品在全年的售价分布情况的基础上，我们将 2019 年副产品销售方案分为“淡季”与“旺季”两个时间区间分别考虑。为简化模型，选择在每周的周末销售副产品。“淡季”的销售原则是只卖超出最大存栏上限的副产品，而到达“旺季”则保证满足存栏下限要求的前提下尽量多销售以实现利润的最大化。最终经模型检验，两个时间区间内的销售方案均可满足货车的运输能力。

模型三用来计算每周的饲料消耗量。在模型二的基础上，根据状态更新函数可得到每天所有存栏猪的数量、状态，并能得到每头猪每天的饲料消耗量，进而求和计算出当天饲料的消耗量，并按周累加最终得到每周的饲料消耗总量。

关键词：状态更新；配种计划；利润最大化；销售方案。

1 问题重述与分析

1.1 问题重述

本题目要解决的中心问题是：为猪场设计连续的生产销售计划，以实现充分利用现有资源，实现利润最大化。目前已知的信息较多，我们将其概括如下：

1>种猪配种信息：

种猪之间的配种方式只有文中所给的 A 系公猪与 B 系母猪、C 系公猪与 D 系母猪及纯系配种这三种情况。其中 A 系公猪与 B 系母猪配种所产出的 F 系母猪可以作为种猪产品，F 系公猪全部将作为副产品销售；C 系公猪与 D 系母猪配种产出的 E 系公猪可作为种猪，E 系母猪全部作为副产品销售；最后纯系配种生产出的猪，母猪可用于生产母猪的更新替换，公猪可作为副产品销售。

2>猪只生产过程：

猪只的生产过程包括配种怀孕、分娩哺乳、保育、育肥与更新等环节。生产母猪在 30 至 32 周龄时配种，怀孕期大约 116 天，分娩率 88%，产活的仔数公母比例大约为 1:1；仔猪的分娩哺乳期为 21 天，在此期间仔猪的死亡率为 10%；仔猪保育至 10 周，保育期的死亡率约为 3%；仔猪育肥至 23 至 24 周，育肥期间的死亡率约为 2%；外售 F 系母猪 15 周时开始选种，选种成功率为 65%，自留母猪 21 周开始测定，自留率约为 65%；E 系公猪 23 周开始选种，选种率约 40%。母猪从怀孕到下一次配种怀孕周期约为 21 周。

3>猪场的产能情况：

猪场目前拥有各类猪舍共 36 件，其中 8 间产房，每间产房可最多可同时养 24 头猪，但由于考虑产房的维护，所以实际按照 6 间产房来计算；14 间保育舍，每间 12 个栏位，每个栏位根据猪体重的不同最多养 18 至 24 头猪，保育期间最终将猪养至 25kg 转至育肥；14 间育肥舍每间 24 个栏位，每个栏位根据猪体重的不同最多可养 23 至 46 头不等。

4>副产品销售情况：

题目中将副产品按照体重分为 4 个类别，附件中给出了 2019 年不同月份对应的副产品售价，同时副产品销售时，其运输车的最高运力有限，共两辆运输车，每辆车每周的最大运载次数为三次。

本文需要我们解决以下三个问题：

- 1.制定每周的配种计划以及生产公猪、母猪的更新计划，并填写附件 8；
- 2.制定四种副产品每周的销售计划，并填写附件 9；
- 3.测算出每周各类饲料的消耗量，并填写附件 10.

1.2 问题分析

1.2.1 问题一分析

问题一要求我们给出猪的配种计划以及生产母猪生产公猪的更新计划，所以我们将第一问拆解为三个部分：

1.考虑销售计划约束，得到初步配种方案

这里首先要明确，在给定配种计划时是按照按天配种还是按周配种或者按月配种。在有初步的配种时间方案后，下一步考虑在配种的日期需要如何配种，考虑配种方案时需要满足两个根本的约束，分别为：1)猪栏存猪数不能低于 9000 头；2)需要满足附件 7 中每个月的销售计划。考虑这两点约束可以得到初步配种方案。

2.考虑对生产公母猪的更新，获得修正的配种计划

题目中明确提出，公猪采精后一年便要 100%更新，母猪从配种起，两年内不可更新，两年以上可有 65%的母猪更新，所以在考虑公母猪更新的基础上，还需要修正上一步得到的初步配种计划。

3.考虑下一步副产品的销售，扩充产能得到最终配种计划

第一部分中我们得到的初步配种方案可以满足种猪每个月的销售计划以及 9000 头猪存栏的下限要求，第二部分我们修正后的配种方案，在第一部分的基础上，考虑了生产猪的更新需求。但是只考虑这两个约束得到的配种方案并没有考虑副产品的销售，没有扩充产能实现产能的最大化，无法获得更多的利润。所以在初步配种方案的基础上，我们还需要尽量多配种实现产能最大化，以便于第二问售卖副产品，获得更多的利润。

1.2.2 问题二分析

问题二是需要在问题一得到最终配种方案的基础上，结合副产品每个月不同的销售价格合理配置销售计划，同时考虑车辆、饲料成本来实现利润的最大化，但在此过程中始终要保持猪的存栏数不能低于最低下限，但同时也不能高于最大存栏量。

1.2.3 问题三分析

问题三需要我们解决的问题是测算每周饲料的消耗量。由于我们所有的过程都考虑是按天更新，所以经过问题一与问题二最终可以得到每天所有猪的体重、日龄以及处于各种类别下猪的数量，所以问题三相对于问题一与问题二而言是一个计算问题，在现有状态的基础上计算出每天的饲料消耗量，进而统计出每周的消耗量即可。

2 符号说明

符 号	说 明
$Age_{(days)}$	日龄
M	体重
N_s	种猪每天的销售总量
N	种猪当月的销售总量
d_{Boars}	生产公猪距离更新的天数
$T_{BoarsReplace}$	公猪须更新日龄
$T_{SowReplace}$	母猪可更新日龄
P_{feed}	哺乳猪数量
P_{cons}	保育猪数量
P_{fatten}	育肥猪数量
P_{pure}	种猪数量
P_{byPro}	副产品猪数量

3 模型建立与求解

3.1 问题一模型建立与求解

3.1.1 模型假设

<1> 题目中给出 A 公猪与 B 母猪配种，得到的一窝 F 母猪约为 11~12 头，且公母比例为 1:1，为了为配种方案的制定留出一定安全空间，这里假设都取下限 11 头；

<2> 假设 A、B、C、D 四类纯系公母猪配种分娩出的一窝哺乳仔猪总数为 20 只，且公猪与母猪的比例为 1:1；

<3> 由于题目中没有给出 2020 年的销售计划，但为了给 2020 年种猪的销售留出一定裕度，所以需要在 2019 年提前为下一年配种，这里我们假设 2020 年的销售计划与 2019 年的销售计划相同。

3.1.2 模型建立与求解

问题一的建模思路流程图为：

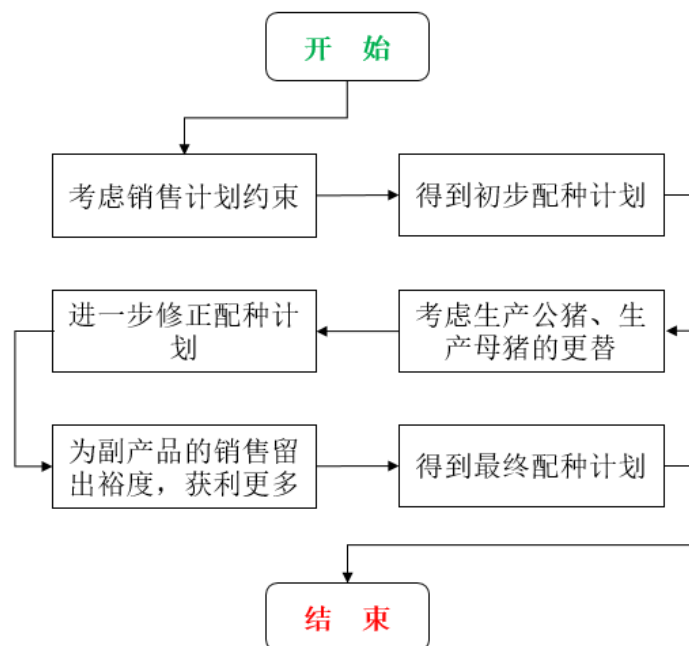


图 3.1 问题一总思路流程图

为辅助我们更好地完成问题一的模型，在解决问题一之前，我们首先需要对题目中的已知信息做更加深入的理解，并对附件中的数据进一步挖掘。

问题一最终需要给出每周的配种计划以及公猪、母猪的更新计划。但在给出配种计划之前，需要以 2019 年 1 月 1 日的初值为基础，由于本题目的已知信息较多，所以在进行配种计划之前，需要结合题目细致分析附件 6 中的初始数据以及其他已知条件。我们知道，所有的配种方式可以分为两个大类，其中第一类是不同类别的公猪母猪杂交，第二大类是纯系猪之间的自交，两种不同类别的公母猪配种方式最终可得到的结果如下：

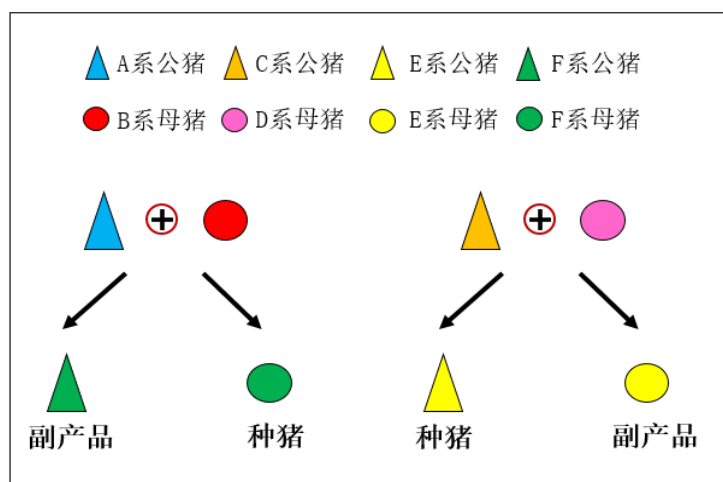


图 3.2 不同类公母猪配种育仔情况

如图 3.2 所示 A 系公猪与 B 系母猪配种可以得到 F 系公猪、母猪，其中公猪是副产品，母猪是种猪；C 系公猪与 D 系母猪交配可得到 E 系公猪、母猪，其中公猪作为种猪，母猪作为副产品。

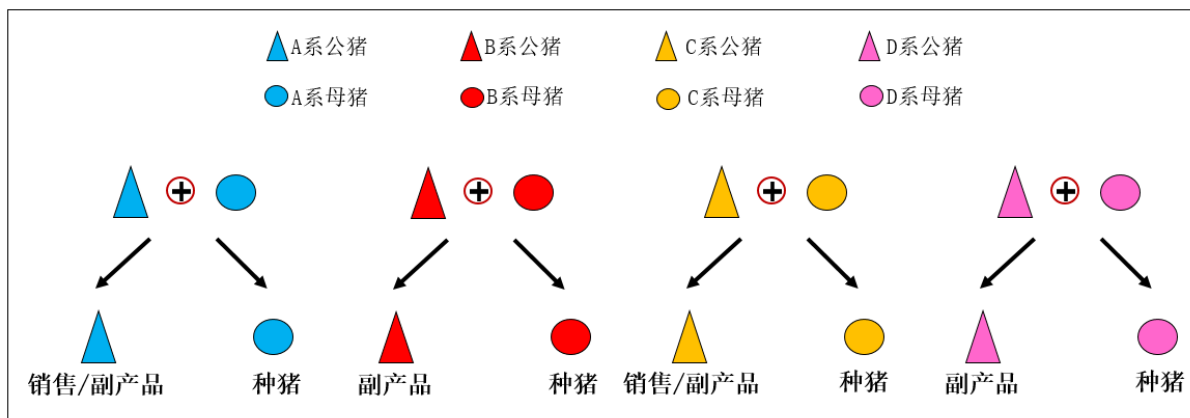


图 3.3 同系公母猪配种育仔情况

如图 3.3 所示，同系公母猪配种共有四种情况，分别为 AA、BB、CC、DD，同系猪配种得到的母猪都为种猪，其可用于自留去替换现有的生产母猪，公猪可以有部分用来自留，也可按照副产品销售。但结合附件 7 给出的销售计划，其中 A、C 两类公猪是有销售需求的，所以这两类公猪可自留，可作为种猪销售，也可按副产品销售。

在了解所有猪配种的类型后，下面需要明确孕育的仔猪在不同的时间段处于什么状态，虽然题目中给出的阶段划分方式如下：

$$Age_{(days)} \in \begin{cases} [0, 21] & \text{(哺乳期)} \\ (21, 70] & \text{(保育期)} \\ (70, 161 \sim 168] & \text{(育肥期)} \end{cases} \quad (1)$$

但在挖掘附件 6 的数据时我们发现实际的阶段划分与题目中给的标准有一些出入，我们将附件 6 中的关键数据挖掘如下：

表 3.1 初始数据中不同种类公母猪保育、育肥时间统计

母/公(days)	A 系	B 系	C 系	D 系	E 系	F 系
开始保育	28/28	28/28	28/28	28/28	26/28	28/无
结束保育	76/75	76/76	76/76	76/76	72/76	76/无
开始育肥	80/80	80/80	80/80	83/81	80/81	80/无

结束育肥	185/185	185/185	185/185	185/185	185/185	184/无
------	---------	---------	---------	---------	---------	-------

将从附件 6 统计的数据反映到图上，如图 3.3 与图 3.4 所示。

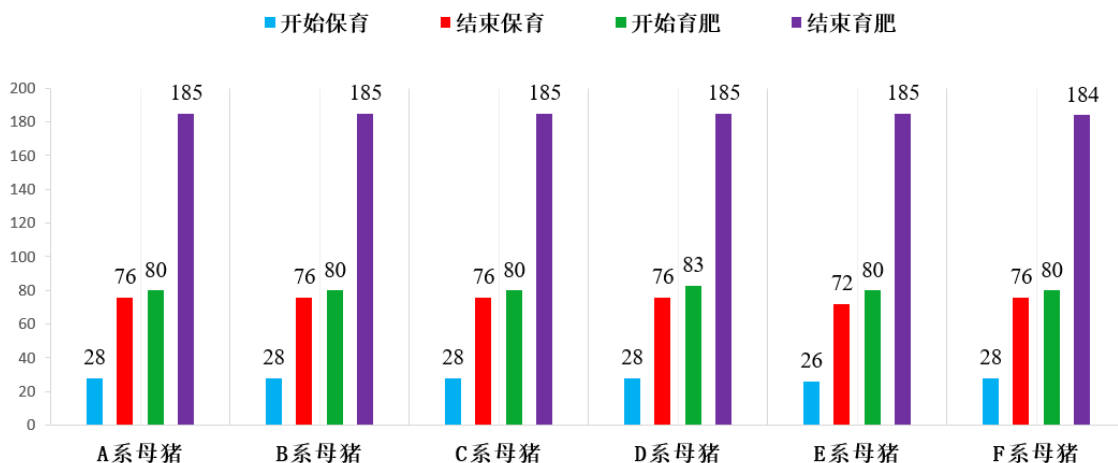


图 3.4 母猪保育、育肥时间统计图

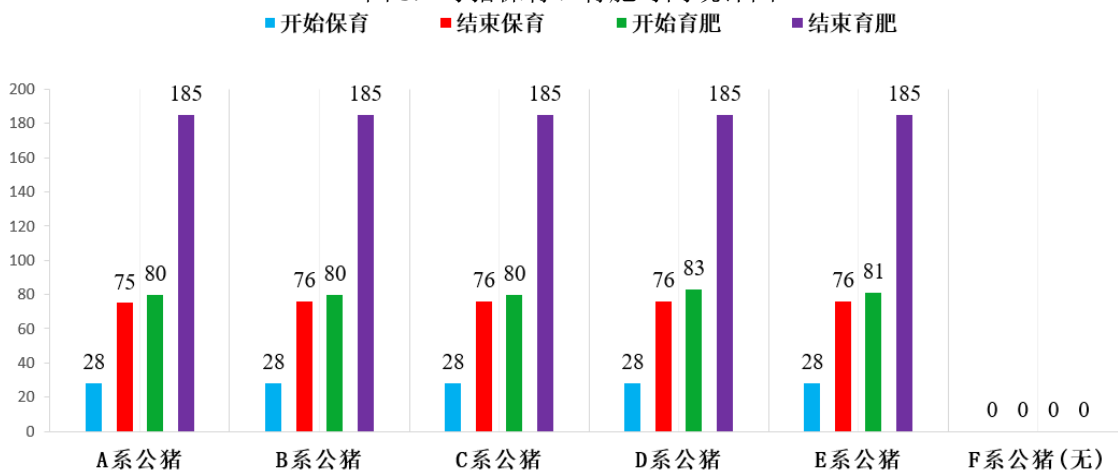


图 3.5 公猪保育、育肥时间统计图

根据图 3.4 和图 3.5 可以明显看出保育的时间跨度以及育肥的时间跨度，区别于题目中给出的不同时间跨度(1)，可以看出实际的哺乳、保育与育肥时间周期为：

$$Age_{(days)} \in \begin{cases} [1, 28] & \text{(哺乳期)} \\ (28, 76] & \text{(保育期)} \\ [80, 185] & \text{(育肥期)} \end{cases} \quad (2)$$

首先可以看出的是当哺乳仔猪的日龄到达 28 天时才进入保育室，结合附件 1 给出的饲喂程序，可以看到饲料 3 的食用日龄是 21~28 天，所以说明将哺乳仔猪转至保育室吃饲料之前，首先还是要要在产房吃 7 天的饲料，这个时间段称为断奶期。而后将仔猪转移至保育室，知道日龄达到 76 天。但从图 3.4 与图 3.5 反映出的数据来看，A、B、C、D、E、F 不论哪种猪的育肥日龄都是从 80/81 开始的，我们在原始数据中保育猪的类别中没有发现日龄为 76~79 天的猪，经过仔细的分析，我们发现只有副产品猪才会将其称为育肥猪，而所有的种猪都是从保育猪直接转变为待售种猪，但种猪由保育猪转变为待售种猪后，只能将其放在育肥舍，所以统计育肥的开始时间时，不能只按照育肥猪来统计，这样只是统计了副产品而没有顾及到所有的种猪，重新修正后的年龄/状态对应关系如下：

$$Age_{(days)} \in \begin{cases} [1, 28] & (\text{哺乳期}) \\ (28, 76] & (\text{保育期}) \\ (76, 185] & (\text{育肥期}) \end{cases} \quad (3)$$

同时题目中提到 F 类的母猪周龄到达 15 周时开始选种，E 系公猪周龄 23 时开始选种，自留母猪 21 周时开始测定。但是附件 6 中，将所有的种猪经过保育后全部称为待售种猪，所以这里分析题意可知，待售种猪的含义并不是已经经过选种等待销售的意思，这里种猪到达 77 天时进入待售状态，其还没有经过选种，也没有经过自留测定，所以当周龄到达 15 周时我们需要对周龄为 15 的母猪选种，到达 21 周时，我们需要对 A、B、C、D 四类母猪测定，选择其中的 65% 自留，剩下 35% 将其转至副产品销售，到达 23 周时，我们首先需要对 A、C 两类公猪选种，选种的百分比为 40%，将其作为最终的待售种猪，剩下 60% 转为副产品销售，对于 B、D 两类公猪，在保留一定自留的基础上，将其全部转为副产品销售。

处理附件 2 中的生长数据时，由于数据中针对每天的体重值，给了不止一个，而且数量不固定，所以首先要通过合理的数据处理方式，得到每天对应的一个体重值，我们的处理方式为：

假设每天给出的体重数量为 k 个，同时其体重值分别为 $M_1, M_2, \dots, M_k$ ，此时我们选择对每个体重值设定合理的权重，最终加权得到当天对应的平均体重。设定权系数时，第 k 个体重对应的权系数 w_k 为：

$$w_k = \frac{M_k}{M_1 + M_2 + \dots + M_k} \quad (4)$$

进而可以得到第 i 天对应的体重值 $M|_i$ 为：

$$M|_i = w_1 M_1 + w_2 M_2 + \dots + w_k M_k \quad (5)$$

在得到每天对应的平均体重后，由于附件数据中还存在一些时间间断，这期间内的体重采样值丢失，所以在得到每天平均体重的基础上需要通过插值方法来得到这些间断处的计算体重。本文我们采用的方法是多项式插值，为了尽量减小在插值过程中引入误差，我们采用五阶多项式来实现多项式插值，具体的插值计算过程如下，假设插值多项式为：

$$P(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5 \quad (6)$$

式中 $a_0, a_1, \dots, a_5$ 为待定系数，通过加权得到的样本点数据可以通过求解方程组的形式来计算待定系数。

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^5 \\ 1 & x_2 & x_2^2 & \dots & x_2^5 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_6 & x_6^2 & \dots & x_6^5 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_5 \end{bmatrix} = \begin{bmatrix} P(x_1) \\ P(x_2) \\ \vdots \\ P(x_6) \end{bmatrix} \quad (7)$$

求解方程组(7)需要 6 个样本数据点，但经处理后的样本远不止 6 个数据点，所以通过不断逼近的方法找到最优的一条五阶多项式插值曲线，经计算我们初步得到的“体重-时间”多项式为：

$$P(x) = -16.26 + 1.472x - 0.03176x^2 + 0.0004041x^3 - 2.026 \times 10^{-6}x^4 + 3.463 \times 10^{-9}x^5 \quad (8)$$

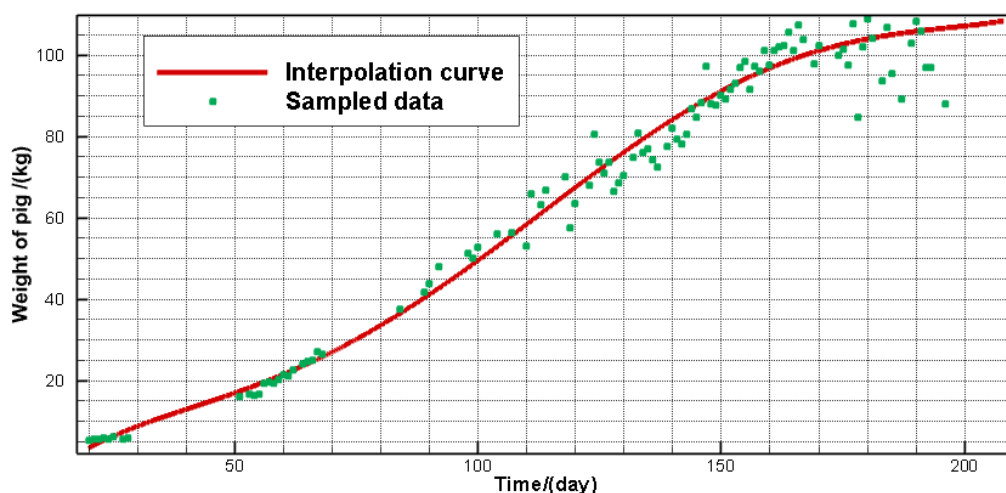


图 3.6 猪的“体重/日龄”计算采样点与插值曲线

这条初步得到的曲线，由于生长数据点有限，我们根据多项式只拟合到了第 208 天，但检查附件 6 的猪只存栏发现，其中很多日龄大于 208 天的猪，尤其部分生产猪，日龄甚至大于 500 天，由于部分生产猪是需要被替换的，而替换后将其作为副产品按千克出售，所以必须还要考虑大于 208 天的体重模型。结合之前的数据分析我们知道，要将猪育肥至 185 天，结合实际情况，育肥期后猪的体重不会再有明显的上涨，而是在一个小的范围内上下波动。此处为了简化模型，我们假定 208 天以后猪的体重保持第 208 天的体重不变，所以最终得到的猪“体重-时间”模型多项式应为：

$$\begin{cases} P(x) = -16.26 + 1.472x - 0.03176x^2 + 0.0004041x^3 \\ \quad - 2.026 \times 10^{-6}x^4 + 3.463 \times 10^{-9}x^5 & (20 \leq x < 208) \\ P(x) = 108.3443 & (x \geq 208) \end{cases} \quad (9)$$

得到分段插值函数的图像为：

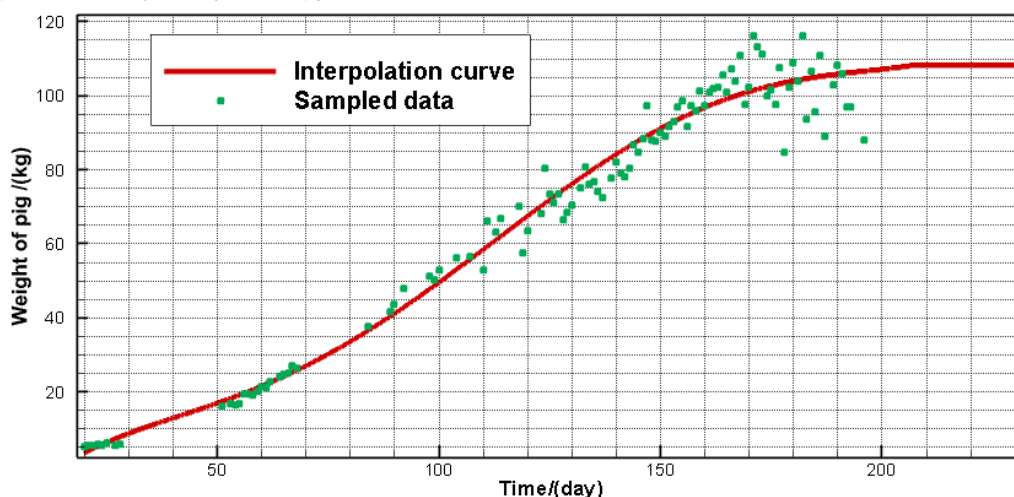


图 3.7 猪的“体重/日龄”计算采样点与最终插值曲线

在建立好这些基础模型的基础上，下面开始细致考虑问题一的模型建立。结合对问题一的分析与图 3.1 的流程图我们可将问题一具体拆解为三个部分，三个部分相互衔接，最终得到题目要求的配种计划以及生产公猪生产母猪更新计划，下面对三个部分分别阐述具体思路过程：

1. 考虑种猪销售计划约束，得到初步配种方案

第一部分的处理思路流程如下：

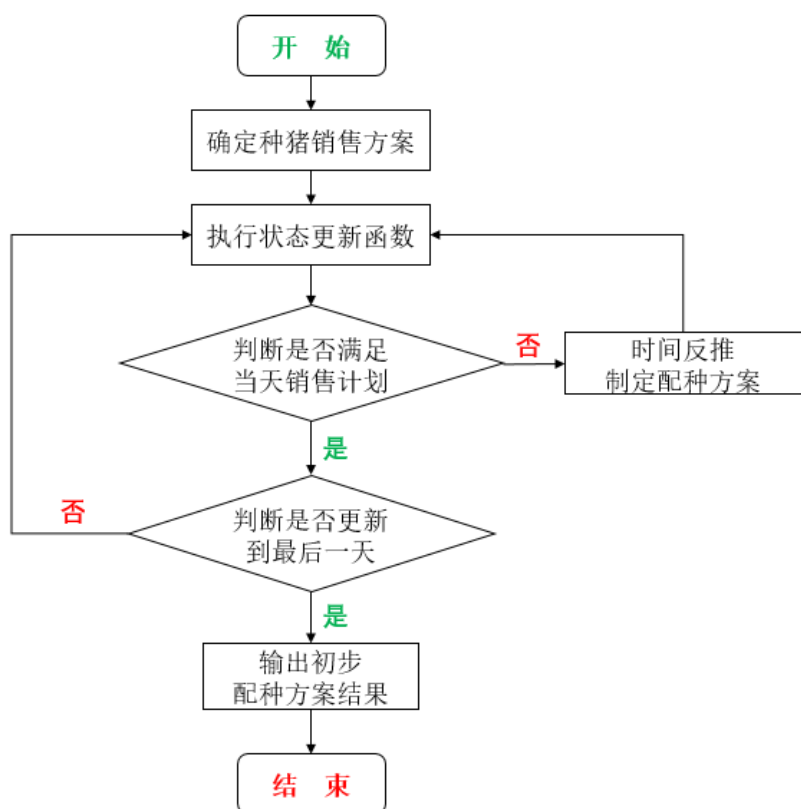


图 3.8 “初步配种方案”的确定流程图

如图 3.8 所示，在确定“初步配种方案”之前首先需要在结合附件 7 “月销售计划”的基础上，明确每个月具体应该如何销售种猪。考虑实际情况，一个养猪场面向的采购商是很多的，并不是只单独面向某一家采购商，所以对于猪场而言，最贴合实际的情况是每天都有猪卖出去的，所以猪的销售在附件 7 的基础上，我们确定的种猪销售方式为——按天销售，并且每天的销量采用加权平均的方式。

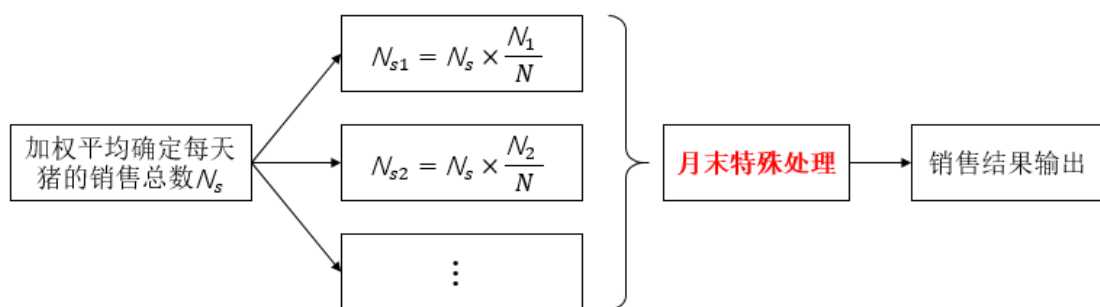


图 3.9 按天销售方法流程示意图

这个过程相对抽象，我们将结合实例重述确定按天销售计划。假设对一月而言，附件 7 给出一月的总销售计划为：

表 3.2 一月种猪销售计划

猪品种	需售数量(头)
C 系公猪	28
E 系公猪	53
F 系母猪	150
合计	231

从表 3.2 中可以看出一月需要售出 231 头猪，同时一月共有 31 天，所以首先可以计算出每天需要售出的头数为：

$$\left\lceil \frac{231}{31} \right\rceil \approx [7.452] = 8 (\text{头}) \quad (10)$$

为了防止到月底出现没有卖完的情况，所以这里在平均值的基础上采用“进1法”来得到每天要卖的头数。在确定每天要卖8头的基础上，可以按照每个类别猪数量占总数百分比来判断每类猪每天需要售出多少头。

$$\begin{cases} C: \left\lceil \frac{28}{231} \times 8 \right\rceil = [0.9697] = 1 (\text{头}) \\ E: \left\lceil \frac{53}{231} \times 8 \right\rceil = [1.8355] = 2 (\text{头}) \\ F: 8 - (1 + 2) = 5 (\text{头}) \end{cases} \quad (11)$$

E系公猪每天卖2头，所以卖到第27天时，这种销售方式将会被改变，此时剩余C系公猪2头，E系公猪1头，F系母猪20头，但本月还有5天，还是要尽量保持按照平均的销售方式售卖不同品种的猪，最终可以汇总得到1月每天的种猪销售需求：

表 3.3 一月种猪按天销售结果汇总

日期 \ 品种	C系公猪(头)	E系公猪(头)	F系母猪(头)	合计(头)
1~26	1	2	5	208
27	1	1	4	6
28	1	0	4	5
29	0	0	4	4
30	0	0	4	4
31	0	0	4	4
合计	28	53	150	231

通过按天销售方法简述结合一月每天的销售方法说明，可看出本方法在分配每天的销量时，考虑了现实情况，具备一定合理性。

在确定每天种猪的销售方案后，下面需要给定每天猪的状态更新函数。状态更新函数的主要目的是为了每天随着猪的日龄增大，猪的状态会发生不同的变化，我们将所有的状态变化梳理如下：

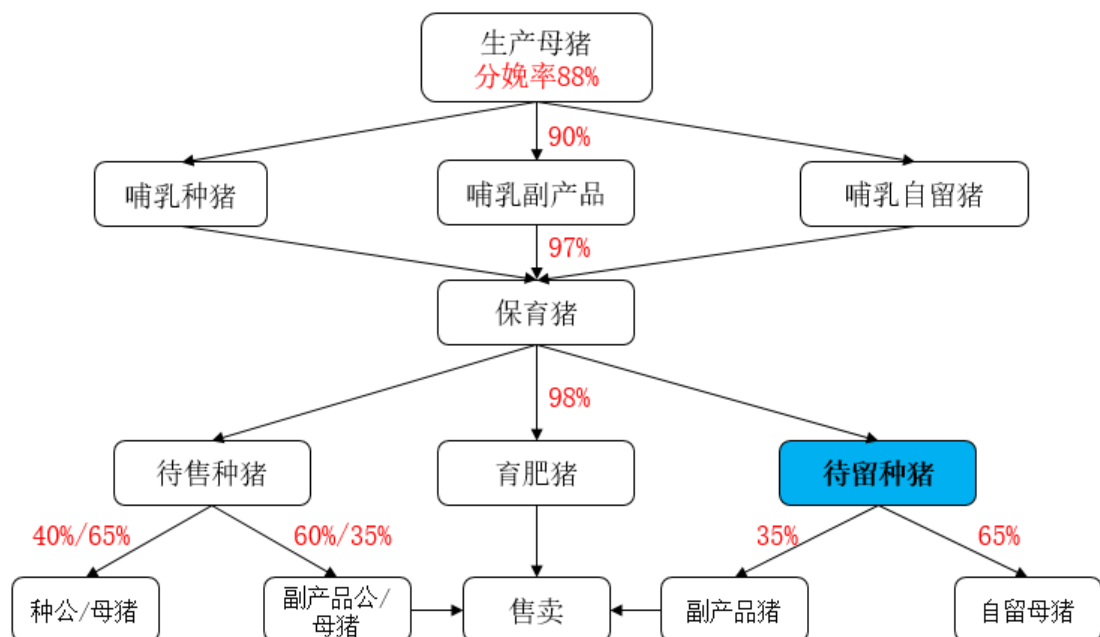


图 3.10 猪的状态变化汇总

从图 3.10 可以看出，生产母猪的分娩率为 88%，成功分娩的仔猪可以分为三类，其中第一类是哺乳种猪、哺乳副产品以及哺乳自留猪，在哺乳期他们的存活率均为 90%。这里哺乳自留猪的含义是我们自己给定的，在后续替换生产母猪时，除了将现存的后备母猪替换完外，还需要借助新生的仔猪，所以我们将用于后续更新生产母猪的仔猪命名为哺乳自留猪，这样区分后有效降低了模型的复杂程度，避免和所有的待售种猪混在一起。自留种猪的出生依赖于我们的配种方案。三类猪都需要保育，保育期间的存活率为 97%，而后三种猪分别进入三种状态，题目中将经过保育期的种猪命名为待售种猪，副产品命名为育肥猪，自留猪命名为待留种猪。待售种猪中，对于所有母猪，当周龄到达 15 周时，其中 65%作为最终选种后的母猪，其余 35%将其转为副产品销售；公种猪周龄到达 23 周时，40%作为最终选种后的公猪，其余 60%用于副产品销售。所有的育肥猪都是当做副产品销售。而最后的待留种猪，周龄到达 21 周时，65%充当后备母猪，剩下的 35%将其作为副产品销售。

下一步的工作便是在状态更新函数的基础上，每天时间推进，推进后检查现存的猪是否满足销售计划约束，如果不满足，则需要时间反推来制定反推当天的配种计划，反推确定配种计划的具体计算过程如下：

假设 2019 年 8 月 1 日当天需要销售 F 系母猪 10 头，但是根据状态更新函数发现 8 月 1 日当天只现存 6 头 F 母猪还差 4 头，所以首先考虑选种时间，以及只有选中后的猪才可以销售的前提下，在 8 月 1 日当天往回推 15 周，也就是 4 月 12 日当天至少需要有 $\left\lceil \frac{4}{65\%} \right\rceil = \lceil 6.1538 \rceil = 7$ (头) F 系母猪出生，但是考虑一对 A 公猪与 B 母猪产出的 F 系母猪一窝有 11~12 头，这里我们按照 11 头验算：

$$\begin{cases} \text{哺乳: } \langle 11 \times 90\% \rangle = \langle 9.9 \rangle = 9 \text{ (头)} \\ \text{保育: } \langle 9 \times 97\% \rangle = \langle 8.73 \rangle = 8 \text{ (头)} \\ \text{育肥: } \langle 8 \times 98\% \rangle = \langle 7.84 \rangle = 7 \text{ (头)} \end{cases} \quad (12)$$

为了保证猪的数量是够的，在处理小数时我们不可再用“进一法”，而是采用“去尾法”，可以看出一窝 F 系母猪出生可刚好满足 7 头的需求。

所以考虑怀孕周期 116 天需要再往回推 116 天配种，从 4 月 12 日往回退 116 天也就是在 2018 年的 12 月 24 日当天需要为一对 A 公猪与 B 母猪配种。但考虑如果仅有 1 对 A 公猪与 B 母猪配种，只能刚好满足 4 月 12 日 7 头的需求，但并没有考虑分娩率 88%，所以此处我们需要配对的公母猪对数为 $\left\lceil \frac{1}{88\%} \right\rceil = \lceil 1.1364 \rceil = 2$ (对)。这样考虑了所有的意外比例，可以保证在 8 月 1 日 F 系母猪的供应是满足需求的。我们在通过反推法制定好配种计划的基础上，需要在新哺乳仔猪出生的当天执行状态更新函数，继续向下一天遍历。

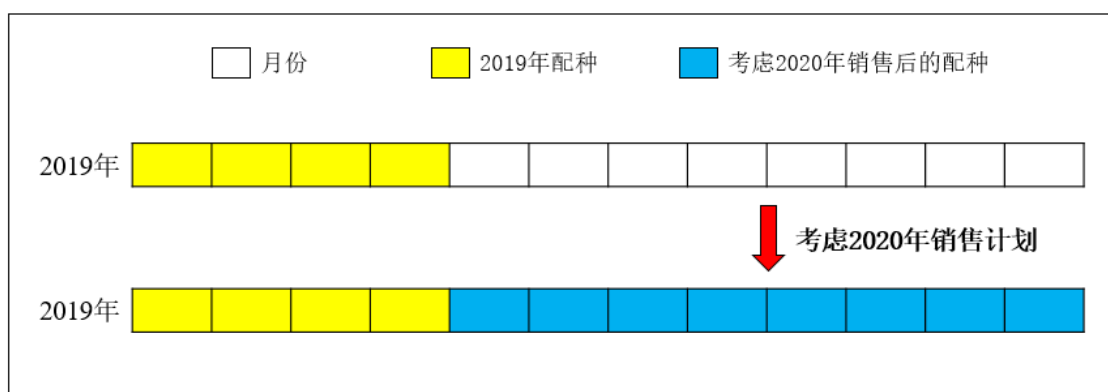


图 3.11 配种时间跨度示意图

但在状态更新函数推进的过程中我们发现 C 系公猪的数量在第一个月就出现了数量不够的情况，此时即使 C 系公猪在 2019 年 1 月 1 日能出生，也无法满足一月的销售计划。由于销售计划是根据订单需求信息以及市场变化提前预测出的，此时出现了初值与销售计划矛盾的情况，所以我们只能在检查的基础上，对销售计划稍作修改。根据修正后的计划来制定初步的配种方案，修正后的销售计划如下：

表 3.4 修正后的 2019 年种猪销售计划

	1月	2月	3月	4月	5月	6月	7月	8月	9月	10月	11月	12月	合计
A 公	0	17	0	68	0	19	31	30	0	30	0	0	195
C 公	11	0	7	0	13	0	0	15	0	0	0	0	46
E 公	53	84	60	61	62	76	76	50	55	101	34	72	784
B 母	0	0	100	0	0	98	0	0	100	0	0	0	298
D 母	0	120	0	0	100	0	0	0	0	0	0	0	220
F 母	150	260	330	330	150	104	250	250	75	57	100	100	2156
合计	214	481	497	459	325	297	357	345	230	188	134	172	3699

在修正完销售计划的基础上，重新执行状态更新函数，来检验当不满足销售计划时通过时间反推增加配种。由于从配种到最终获得可以销售的哺乳仔猪，其时间跨度都很长，尤其是公猪，时间跨度相比母猪更长

$$\begin{cases} \text{母猪: } 15 \times 7 + 116 = 221(\text{天}) \\ \text{公猪: } 23 \times 7 + 116 = 277(\text{天}) \end{cases} \quad (13)$$

时间大约在 7~9 个月，所以在 2019 年前 7 个月，当发现待销售猪数量不足时，我们是在推算 2018 年的配种计划，只有当到达 2019 年 7 月以后，发现待销售猪不足时，才会获得当年年初附近的配种计划，由于我们不知道 2020 年前 7 个月的销售计划，所以无法直接获得 2019 年后 7 个月的配种计划，但如果不配后 7 个月明显是不合实际的，虽然可以满足 2019 年种猪的销售计划，但无法保证下一年的销售。如图 3.11 所示，为了简化问题，结合假设，2020 年，我们采取和 2019 年相同的种猪销售方案，所以在考虑 2020 年销售计划的基础上，可以获得 2019 年全年的初步配种方案，这样为下一年种猪的销售留出了裕度。综上，经过这些过程，我们首先得到了初步配种方案。

2. 考虑对生产公母猪的更新，获得修正的配种计划

生产公猪母猪的更新计划思路图如下：

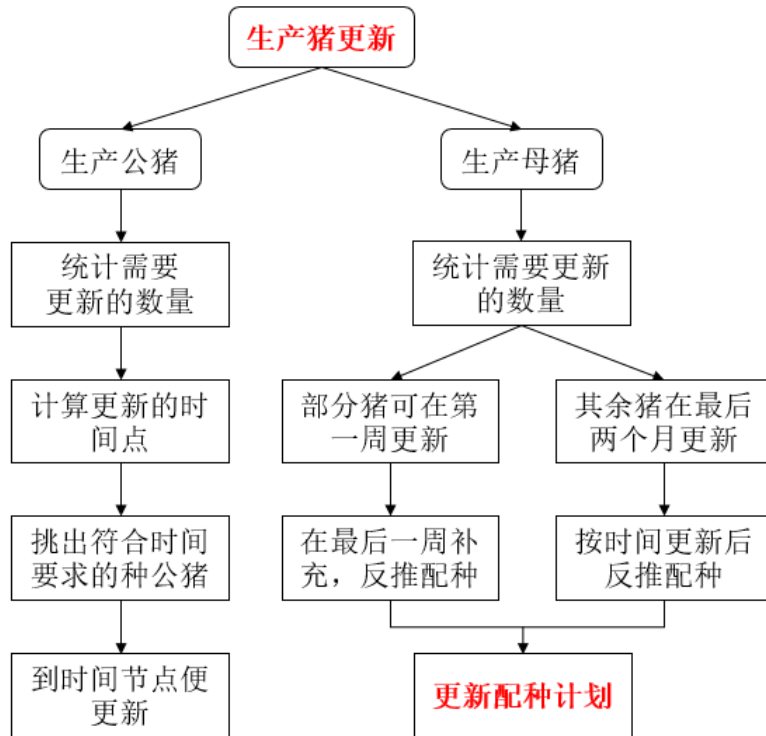


图 3.12 生产猪更新与配种计划修正思路图

考虑猪的更新时，我们将猪分为公猪与母猪分别更新。首先考虑公猪的更新，考察初始 9700 头存栏猪的状态，其中共有 30 头生产公猪，我们知道生产公猪当经过第一次采精后一年便需要更换，将公猪的更新时间换算为天数即当生产公猪的日龄到达：

$$T_{BoarsReplace} = (43 + 52) \times 7 = 665 \text{ (天)} \quad (14)$$

时便需要更新。30 头生产公猪的数量远小于生产母猪，所以我们可利用现存的公种猪作为后备公猪。并且生产公猪的日龄是不同的，所以其更新时间不同，所以首先我们要得到这 30 头猪分别需要过多久更新。假设某一头生产公猪需要过 d_{Boars} 天更新，则我们需要在此时挑选的公猪日龄要求为：

$$Age_{(days)} \geq 43 \times 7 - (52 \times 7 - d_{Boars}) \quad (15)$$

在所有满足日龄要求的公猪中挑选一头年龄最小的作为后备公猪，挑选年龄最小的原因是，当到达更换日时，后备公猪的周龄更加接近 43 周，此时猪的生产活力也是最高的。综上，我们对生产公猪的更换是按天更新，并且从现有的 A、B、C、D 四类公猪中挑选即可，生产公猪的更新不影响配种计划。

在考虑对生产母猪的更新时，由于母猪的生产种猪数量较大，经统计可得母猪总共的生产种猪数量为 1120 头，判断母猪在今年是否会达到更换日龄的计算方法为：

当母猪周龄到达 32 时可以开始配种，在 32 周龄的基础上，再过两年可考虑更换，即日龄达到：

$$T_{SowReplace} = (32 + 52 \times 2) \times 7 = 952 \text{ (天)} \quad (16)$$

时可更新其中的 65%，统计附件 6 给出的猪现存情况可知，在 2019 年内可考虑更换的母猪数量为 514 头，乘以 65% 可得 $[514 \times 65\%] = 335$ (头)，可得今年内需要更换的生产母猪数量为 335 头。其中各类母猪的汇总情况如表 3.6 所示。

进一步挖掘附件 6 中的数据，可知目前现存的备选母猪数量为 54 头，所以可以根据这 54 头优先找到 A、B、C、D 四类母猪中最急需更新的母猪即日龄最大的，将其在第一周用这 54 头去更新，54 头备选母猪的统计结果如表 3.5 所示。

表 3.5 备选母猪数量与急需更新数量统计

	A 系母猪(头)	B 系母猪(头)	C 系母猪(头)	D 系母猪(头)	合计(头)
备选母猪数	7	24	7	16	54
第一周最大可更新数	7	24	2	16	49

表 3.6 生产母猪可更换与需要更换数量统计

	A 系母猪(头)	B 系母猪(头)	C 系母猪(头)	D 系母猪(头)	合计(头)
可更换数量	112	299	5	98	514
需更换数量	73	194	4	64	335
一周后还需更新数量	66	170	2	48	286

从表 3.5 可以看出四类备选母猪每一类的数量，但结合表 3.5 与表 3.6 可以发现 C 系母猪 2019 年总共只更新 4 头，其中在第一周可更新的数量的只有两头，所以此时我们备选母猪多出来 5 头，这 5 头直接将其转移至副产品销售即可。提前更新好 49 头后，今年内还需要更新的母猪数变为 $335 - 49 = 286$ (头)，其中每一类猪需要更换的数量见表 3.6。

在确定需要更新的数量后，下一步考虑要更新的时间节点，由于 49 头是我们在第一周就要更新的，所以我们采用按天更新的方式，具体方式与确定销售计划的方式相同，保证每天四类猪按照平均值来更新。对于剩下的 286 头猪，题目中说的非常详细，生产母猪日龄达到 952 天这个时限后，可以考虑按照 65% 的比例来更新，但也不是一定要更新，所以我们选择在 2019 年的最后两个月集中更新这 286 头生产母猪，首先还是需要确定后两个月每天各类母猪的更新数量，确定每天更新数量的过程与确定日销售量的方式完全一致；其次由于此时已经没有备选母猪，所以需要利用状态更新模型，在考虑不同阶段死亡率的前提下来反推之前的配种情况，以保证满足当下的生产母猪更新需求。

在执行完生产母猪生产公猪更新的基础上，我们进一步考虑，到达年底，我们已经用完了所有的备选母猪，包括之前配好种的生产母猪到现在也已经用完了。这样会出现下一年刚开始没有备选母猪可用的情况，所以我们在配完这些备选母猪的基础上，还需要考虑多配一部分后备母猪为下一年生产母猪的更换留出一定裕度。

我们考虑的补充方式为：补充的数量与附件 6 中给出的初始备选母猪数量相同，由于这些猪在 2019 年 1 月的第一个星期就已被全部消耗，所以我们在最后一周保证补齐这 54 头备选母猪。需要补齐的 A、B、C、D 四类备选母猪的数量分布与初始的 54 头完全一致，并且其年龄都是在 32 周左右，更加年轻，更有利于下一年生产母猪的更替。所以此时在给定最后一周每类备用母猪补充计划的基础上，还需要反推 $32 \times 7 + 116 = 340$ (天) 来进一步更新配种方案。综上，经过第二部分后，可以得到初步修正的配种方案，以及题目所需的生产公猪、母猪更新计划。

3.考虑下一步有更多副产品可卖增加产能，得到最终配种计划

前面两部分得到的配种计划只是满足了最低的销售计划以及生产更新计划这两个最低的约束，但其实按照前两步得到的配种方案只考虑了种猪的销售，并没有考虑副产品的销售，实际过程副产品的销售与种猪的销售是一个连贯的过程，所以此时猪舍中还是有很多空位的，如果仅采用初步修正后的配种方案无法进一步增加猪场的获利，所以最后这一步，我们要在之前的基础上，尽可能实现产能最大化。

根据附件 3 的产能数据我们发现，产房最多可容纳的哺乳仔猪数量为 1764 头，但我们知道哺乳仔猪日龄在 28 天以内，都会被强行放在产房中。所以可以初步计算哺乳仔猪每天出生数量上限：

$$\left\langle \frac{1764}{28} \right\rangle = \langle 63 \rangle = 63 (\text{头}) \quad (17)$$

每窝哺乳仔猪的总数大约为 20 头，所以可以计算出每天最多出生哺乳仔猪的窝数为：

$$\left\langle \frac{63}{20} \right\rangle = \langle 3.15 \rangle = 3 (\text{窝}) \quad (18)$$

所以如果每天出生的哺乳仔猪大于 3 窝，会出现最终产房无法容纳的现象，所以初步确定了每天最多只能有 3 窝哺乳仔猪出生。同时检验产房的空位，每天有 3 窝哺乳仔猪出生，所以每天有 3 头生产母猪进入产房，由于每只生产母猪在产房待的时间为 24 天，所以最多占用产房的生产母猪数量为 $24 \times 3 = 72$ (头) 但是产房中共有 144 个空位，所以产房始终是有空位的。同时初步检验货车的运输能力，即是否能够每天运输走 60 头猪。通过附件 5 中给出的“运输能力”我们可以看到，两辆货车每周各可以运输 3 次，其中货车 1 一次可以运输 102 头重量为 100kg 的猪，所以通过合理的设计运输方案，可以满足每天 80 头的运输需求，并且这 60 头猪中并不是所有猪是副产品，还有一部分是待销售的种猪，其并不需要货车运输。综上，我们确定扩充副产品的方案是保证每天都有 3 窝猪出生。从图 3.13 可以看出，在初步生产计划与考虑生产公猪母猪更替的基础上进一步将每天所有出生猪的数量扩充为 3 窝，经过 116 天怀孕期的时间反推，便可得到 116 天之前的 3 对配种数量。

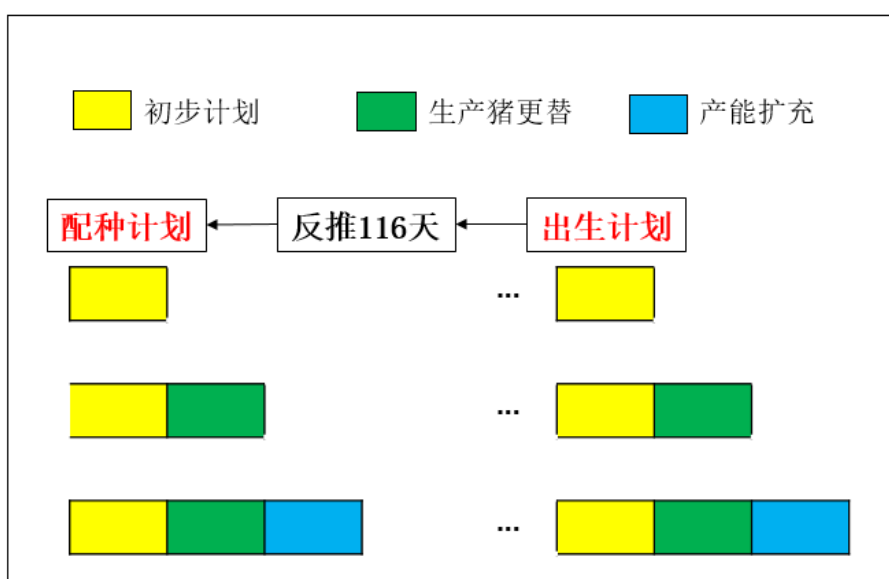


图 3.13 问题一每天配种数量示意图

下一步在得到扩充方案的基础上，需要时间反推来得到配种计划。由于我们的配种目的是增加猪的数量，所以经过前面的分析，在窝数不能增加的基础上我们想让哺乳仔猪一窝的数量最多，所以争取多配 F 母猪与 F 公猪，这样一窝猪的数量为 22~24 头，可保证数量最多。当可怀孕的 B 系母猪都配完后，为简化模型，其他的配种均考虑 A、C、D 三个同系公母猪的配种。完成最终配种这一步后便得到了最终的配种计划，我们将最终配种计划结果与生产公猪、母猪的更新方案均放于附件中，其命名分别为“问题一配种计划.xls”与“问题一生产猪更新计划.xlsx”。

3.1.3 模型检验

模型一首先考虑种猪销售计划得到“初步配种计划”，其次考虑生产公猪与母猪替换得到“初步修正配种计划”，最后考虑实现产能最大化得到最终的配种方案，其满足题目中设定的要求，并为后续副产品的销售在产能最大化的基础上留了充分的裕度。

我们将最终的配种计划作为模型的参数输入，来分别检验是否满足销售方案、生产公猪母猪的更换是否符合要求，并且给定的配种方案能否满足生产猪的更新方案，最后检验保证每天有 3 窝仔猪出生是否会超过猪舍的最大存栏量。

1.配种方案是否满足种猪销售方案

我们在模型建立的过程中，首先得到的初步方案就是基于种猪销售方案制定的，但在建模的开始，我们发现有些销售计划，即使在 1 月 1 日就让仔猪出生，也是无法满足的，同时由于 2019 年的销售计划是在订单需求以及市场变化基础上提前预测的，与真实情况会有一些出入，所以我们在此基础上对种猪的销售计划表稍作处理，最终使得初步配种方案可以满足修正后的 2019 年种猪销售计划。后续对配种计划的修正都是在初步配种计划的基础上添加更多的仔猪，所以模型可以满足种猪销售方案。

2.生产猪更新是否符合题目要求

已知，2019 年需要更新的公猪为 30 头，需要更新的母猪为 335 头，考虑更新公猪时，由于公猪需要更新的数量极少，所以可以直接从现有的公猪中更换，但是母猪要更换的数目较大，我们建模时，将其分为两部分更新，第一部分是用现有的后备母猪直接更新，这部分更新了 49 头，剩下的 286 头需要提前配种，我们将其称为“待留母猪”。所以生产猪的更新是符合题目要求的。

3.配种方案是否满足生产猪更新需求

在得到生产公猪生产母猪更新方案的基础上，由于生产母猪有 286 头是我们要提前配种的，并且我们考虑了配种过程中的分娩率，哺乳、保育、育肥期间的死亡率以及 65% 的自留率，最终反算出了配种计划，将得到的配种计划，我们称为初步修正的配种计划。由于后续产能的扩充是在初步修正配种计划的基础上增添更多额仔猪，所以模型是符合生产猪更新需求的。

4.扩充后的配种计划是否会超过最大存栏上限

模型确定的配种计划是将每天出生的仔猪数量控制在 3 窝，我们知道产房最多可以容纳的哺乳仔猪数量为 1764 头，且日龄在 28 天以内均要待在产房，所以每天仔猪的最大出生数量是有上限的：

$$\left\langle \frac{1764}{28} \right\rangle = \langle 63 \rangle = 63 (\text{头}) \quad (19)$$

由于仔猪都是按窝出生的，一窝仔猪的数量大约在 20 只左右，所以显然每天最多出生的仔猪窝数为 3 窝。同时检查产房情况，每天会有 3 头生产母猪进入产房，每头生产母猪在产房待 24 天，所以考虑按照我们制定的扩充方案，产房中最多的生产母猪的个数为：

$$24 \times 3 = 72 < 144 (\text{头}) \quad (20)$$

不会超过产房的存栏上限，所以我们制定的最终配种计划不会超过产房的最大存栏量。

表 3.7 问题一模型检验结果

待检验项	配种方案是否满足种猪销售方案	生产猪更新是否符合题目要求	是否满足生产猪更新需求	扩充后的配种计划是否超过存栏上限
检验结果	满足销售方案	符合要求	满足生产需求	未超过存栏上限

由表 3.7 统计得到的结果可以判定问题一建立的模型保证了最低销售需求的要求，以及生产猪的更新需求，并且在保证不会超过最大存栏量大的基础上为副产品的销售留出了足够的裕度。所以可以判断模型具备一定的合理性与有效性。

3.2 问题二模型建立与求解

3.2.1 模型假设

<1> 在销售过程中，由于猪的出生与数量的更新我们都是按天为单位计算的，所以为了合理简化模型，我们假设卡车运输期间没有新仔猪的出生，也没有种猪被出售。

<2> 考虑保育期的仔猪体重都低于 6kg，这种状态下的仔猪其能获得的利润较低，并且考虑实际情况，一般也不会销售处于哺乳期的仔猪，所以这里假设不销售哺乳期的仔猪。

3.2.2 模型建立与求解

问题二是要在问题一得到配种计划的基础上来给定副产品的销售计划，附件 4 给出了一年内四种副产品每个月的价格，分析数据，我们可以看出一年内 4~9 月副产品的销售价格较高，我们将其称为“旺季”；1~3 月与 10~12 月副产品的价格相比 4~9 月较低，我们将其称为“淡季”。显然，在“淡季”，我们要保证猪存栏数不超过上限的基础上，尽量少卖，而“旺季”，我们要保证存栏猪不低于下限的基础上尽量多卖，但是不管哪种情况，我们都是选择在周末销售副产品。下面我们把问题二拆解为两个部分，分开来考虑。但需要注意的一点是，不论是“淡季”的销售计划还是“旺季”销售计划的制定都需要将问题一中的状态更新函数进行更新，问题一给出的状态更新函数只考虑了种猪的销售情况，这里在销售副产品时需要考虑副产品销售后状态的更新问题。

1. “淡季”副产品销售计划

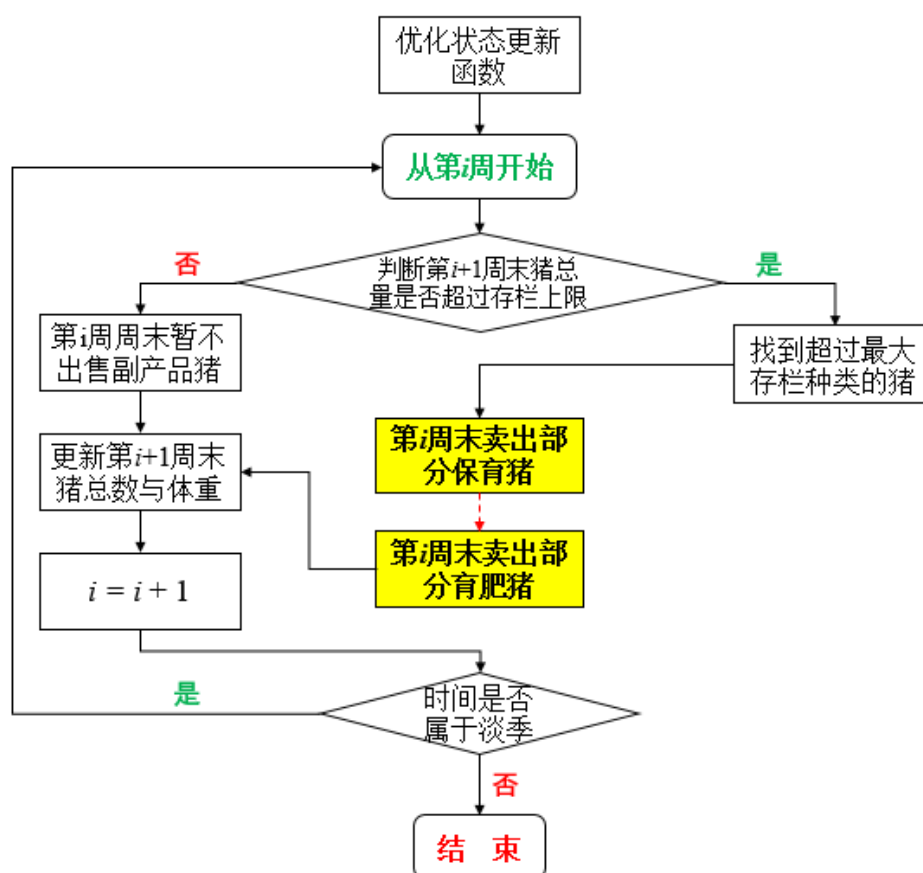


图 3.14 “淡季”副产品销售流程图

流程图中底色标黄的部分是模型的核心，我们将其简要拆分为以下三个具体过程：

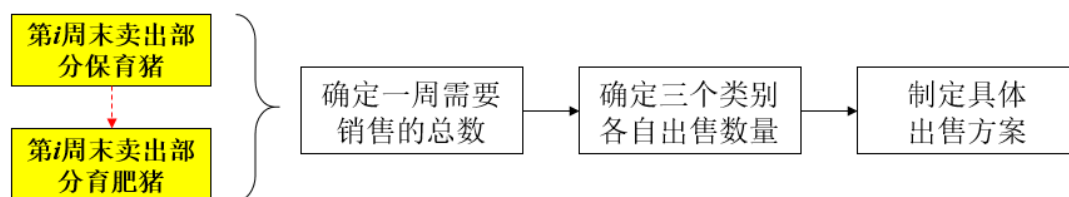


图 3.15 副产品销售计划制定步骤

“淡季”时，我们的目标是在不超过存栏上限的情况下，尽量多存一些猪，将其留到“旺季”时期多销售，销售的具体细节如图 3.15 所示，我们选择销售副产品的时间节点是在周末统一销售。从第一周开始根据问题一的配种计划，可以得到每天所有猪的总数，所以首先判断第二周周末的猪总数是否超过了猪的存栏上限，但此处的存栏上限不是总数上限，而是分开考虑产房、保育室与育肥室的最大存栏数，我们可参照附件 3 给出的每种类别最大存栏数：

表 3.8 不同类别猪的最大存栏量

不同类别的猪	最大存栏量(头)
生产公猪	40
后备母猪	110
生产母猪	1020
哺乳仔猪	1764
保育猪	2675
育肥猪	5657
合计	11266

其中生产母猪的最大存栏数与附件 6 给出的初始状态有矛盾，初始状态下的 9700 头猪中共有 1120 头生产母猪，考虑附件 3 给出的最大存栏数据是一组实验情况，考虑实际情况与题目给定的现实数据，我们选择修改不同类别猪的最大存栏量。由于生产母猪也是放在育肥室的，所以在对生产母猪上限增加 100 的情况下，应该将育肥猪的最大存栏数减去 100，修改后的最大存栏量数据如下：

表 3.9 修改后不同类别猪的最大存栏量

不同类别的猪	最大存栏量(头)
生产公猪	40
后备母猪	110
生产母猪	1120
哺乳仔猪	1764
保育猪	2675
育肥猪	5557
合计	11266

由于不售卖哺乳期的猪，所以如果第二周周末保育猪与育肥猪的现有存栏量未超过最大存栏上限，则第一周周末暂不销售副产品；如果第二周周末的保育猪与育肥猪若超过各自的最大上限，则在第一周的周末卖掉第二周各类超过最大存栏数量的猪即可，这样做保证猪的存栏数都是在上限附近，囤积的猪可在旺季多销售以获得更多的利润。

假设第二周末，保育猪的现有数量为 P_{cons} ，育肥猪的现有数量为 P_{fatten} ，所以可以得到第二种周末两类猪分别超过最大存栏上限的数量为：

$$\begin{cases} \text{保育猪: } N_{\text{cons}} = P_{\text{cons}} - 2675 \\ \text{育肥猪: } N_{\text{fatten}} = P_{\text{fatten}} - 5557 \end{cases} \quad (21)$$

所以第一周周末分别需要卖出的两种类别副产品数量分别为 N_{cons} 、 N_{fatten} 。在确定要卖出的副产品数量基础上，下面在考虑在淡季时，到底应该出售哪些副产品。

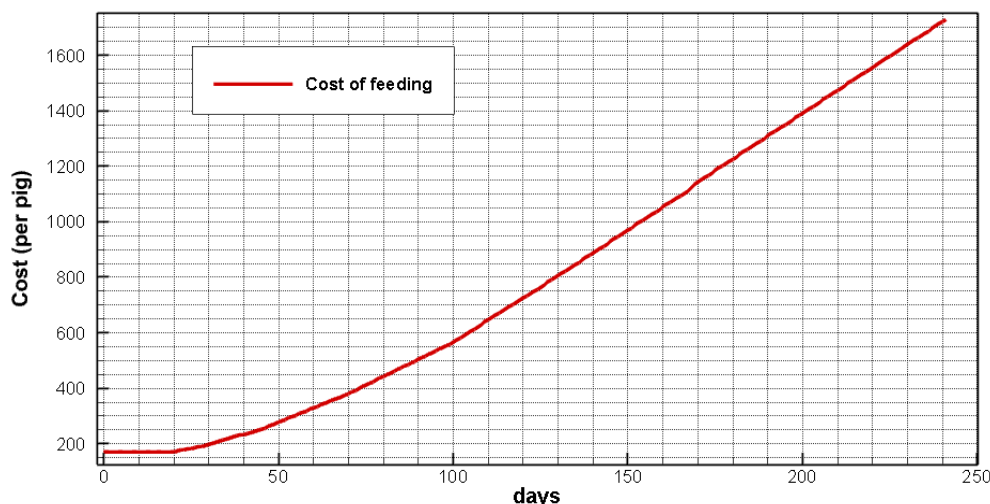


图 3.16 一头猪在不同日龄下的饲料成本

如图 3.16 可以明显看出，随着猪日龄的不断增长，其累计消耗成本也会逐渐增加。在淡季考虑具体需要销售哪些猪时，需要遵循的基本方法如图 3.17 所示。

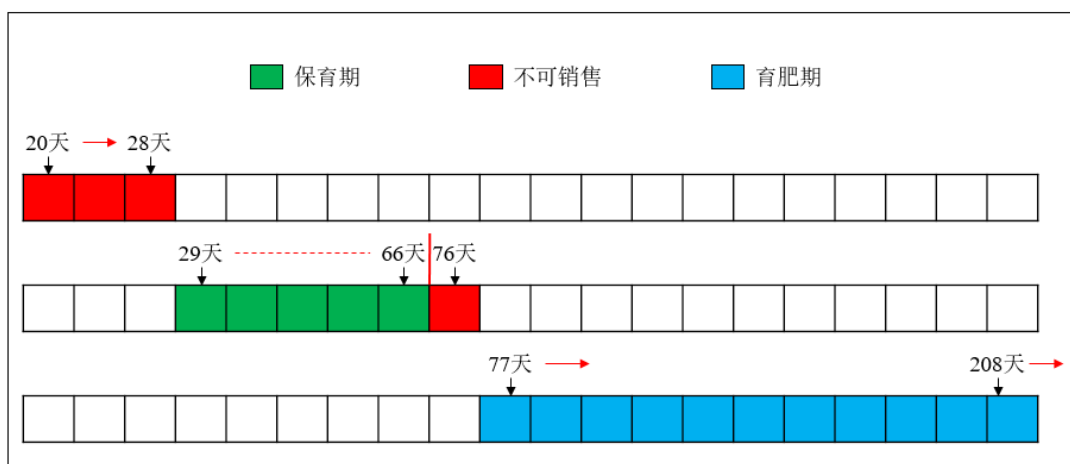


图 3.17 “淡季”阶段销售副产品的选定方法

由假设<2>我们知道副产品销售时是不卖哺乳猪的，可以通过验算来说明在一定程度上假设的合理性。假设某头哺乳仔猪的日龄为 20 天，此时这头猪的成本只有分担生产公猪、生产母猪的成本 171 元，通过模型一得到的“体重/日龄”曲线可以计算出其体重为 5.18(kg)，假设在一月，可以计算出这头哺乳仔猪产生的利润为：

$$5.18 \times 54.18 - 171 = 109.65 \text{ (元)} \quad (22)$$

但如果是一头育肥猪，假设其体重为 100(kg)，其对应的日龄约为 180 天，通过附件 1 可知 180 天对应的成本为 1225 元，此时可以计算得到这头猪的利润为：

$$100 \times 16.45 - 1225 = 420 \text{ (元)} \quad (23)$$

从(22)和(23)这两个结果可以明显看出育肥期的猪其利润更大，虽然这些猪到后期的累计成本较高，但它们的体重更大，所以对单头猪而言，体重相对较大的猪其利润更

大。这为下一步的销售奠定了思路基础，同时也验证了假设<2>的合理性，哺乳期的猪不销售而是等到其进入保育阶段后再考虑售卖。

经过初步的成本检测后，下一步考虑副产品中保育猪的销售思路。我们知道处于保育期的猪将其分为两段，前一段体重大约为 6~20kg，这个体重区间内的猪是可以销售的，其对应的生长时间为 29~66 天；后一段体重大约为 20~25kg，根据题目中的要求，这个区间内的副产品猪是无法销售的，所以我们初步考虑的方案是：在“淡季”如果保育猪数量超标，应该从日龄为 29 天的猪开始售卖，其他的猪待其到达“旺季”统一售卖，这种方法存在一个较大的问题，如果将保育猪按照日龄从小到大售卖，则会出现“截断”造成哺乳猪刚进入保育期便被出售，但与此同时，保育猪中每天可能都有猪进入育肥期，进而造成保育猪的更新赶不上消耗，最终导致保育期的猪越来越少。所以我们考虑采取一种新的方式来避免出现“截断现象”。这里采用均匀售卖的方法，现将思路叙述如下：

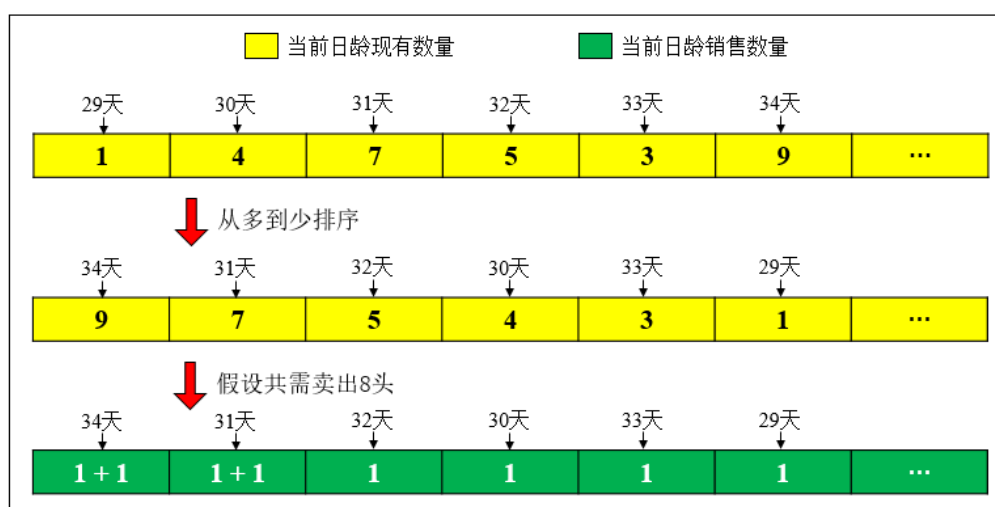


图 3.18 按照排序平均出售思路示意图

如图 3.18 所示，假设截取保育期 29~34 天的时间区间，首先根据状态更新函数可以得到日龄在 29~34 天每个日龄对应的保育猪数量，将这些保育猪数量从高到低排序后，假设要出售 8 头，则按照排序后的结果依次先取一头出售，剩下的两头又从数量最多的开始，再依次取，直到数量到达 8 头为止。采取这种方式来确定要被出售的保育猪，是一种均匀的过程，不会出现之前提到的“截断”的现象，并且日龄对应现有数量较大的猪会被多卖一部分，有效实现了区间内各种日龄猪数量的均匀化。

最后是育肥猪，结合问题一中的生长曲线，我们知道当猪过了育肥期后其体重不会再出现明显的上涨，而是起伏不定，所以在“淡季”如果育肥猪超过最大存栏量，应该首先卖掉日龄大于 208 天的猪，因为这些猪留到旺季体重不会增长而且消耗更多饲料，在卖完这些猪的基础上，考虑育肥猪没有和后一个阶段的过度，所以不需要采用均匀售卖的方式，即可以考虑从育肥期刚开始的猪开始售卖，其他日龄的猪同样留到旺季销售。

第一段淡季我们采用的是囤积副产品猪的销售方式，在第二段淡季：10~12 月，我们采用的方式完全一致，其目的是在经过旺季后，保证副产品猪的数量逐渐恢复到初始状态来实现猪场的连续盈利，同时为下一年副产品的销售计划制定留出足够的裕度。

2. “旺季”副产品销售计划

“旺季”副产品的销售计划如图 3.19 所示

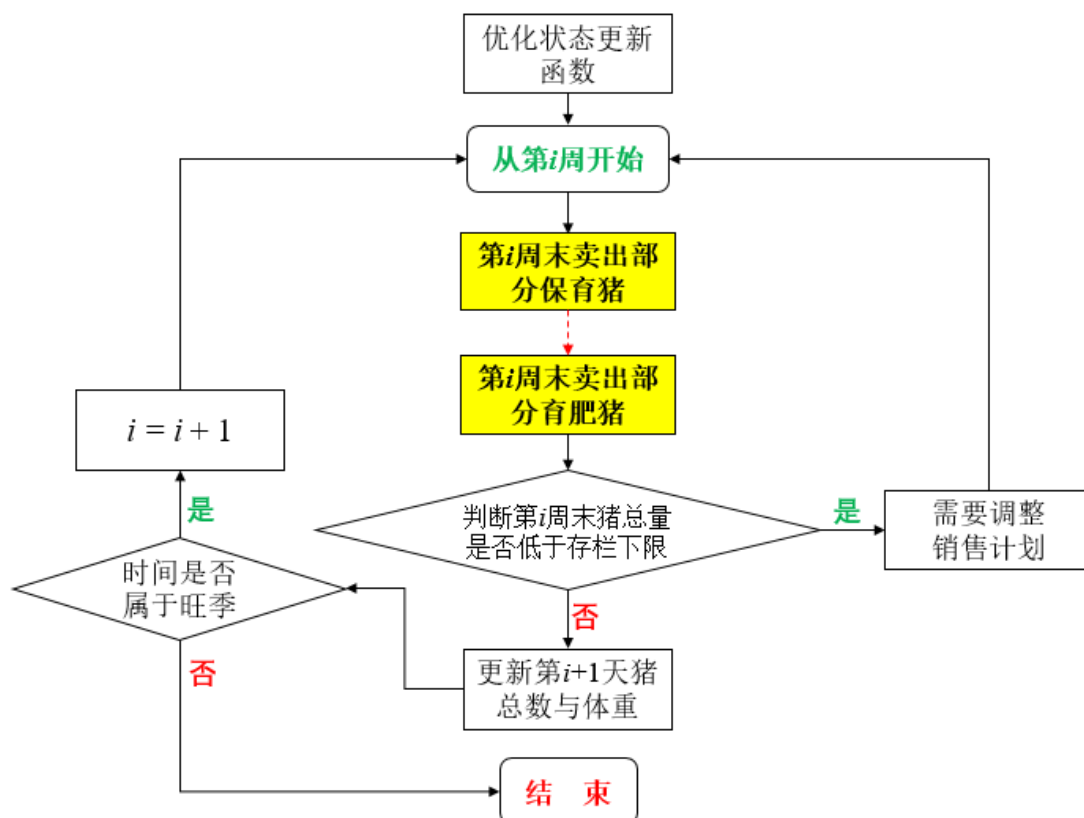


图 3.19 “旺季”副产品销售计划思路图

同样，流程图中标记黄底的部分是模型的核心，其基本过程与图 3.15 完全一致，只是在每个部分采用的方法不同。副产品销售“旺季”时，我们的销售主思路是：在满足题目要求的存栏猪总数不低于 9000 下限的基础上，尽可能多去卖猪，而且优先售卖获利更高的猪，我们的销售策略还是每周选在周末当天销售，但此时与淡季销售策略不同的是，从第一周周末开始由于“淡季”囤积了很多没有销售的副产品，所以可以直接销售，但销售的过程要保持猪的存栏数不能低于存栏下限。明确基本思路后，下一步是确定副产品的具体销售方式。

首先我们知道经过“淡季”副产品猪的囤积，“旺季”第一周周末三类猪的现有存栏数是处于饱和状态的。所以旺季第一周周末哺乳、保育、育肥猪的现存数量如表 3.9 所示。由于此时存栏猪的总数为 11266，为了防止意外情况的出现，我们在题目给定存栏猪下限为 9000 的基础上，将其提高为 9166，所以第一周允许我们售卖的副产品个数为 2100 头。按照两类猪的现存数量来按比例划分各自可售卖的个数，三类猪的现存总数为：

$$P_{all} = P_{cons} + P_{fatten} \quad (24)$$

带入数据可得 $P_{all} = 2675 + 5557 = 8232$ (头) 按照比例划分可得每种类别在第一周可售卖的数量如下：

$$\begin{cases} \text{保育猪: } \left\lceil 2100 \times \frac{2675}{8232} \right\rceil = \lceil 682.39 \rceil = 683 \text{ (头)} \\ \text{育肥猪: } 2100 - 683 = 1417 \text{ (头)} \end{cases} \quad (25)$$

但是货车是否足以支撑 2100 头的数量，需要经过核算，货车的运力如表 3.10 所示：

表 3.10 货车运力汇总与安全重量计算结果

	100(kg)	50(kg)	15(kg)	运输最低重量
车 1(头)	102	200	600	9000(kg)

车 2(头)	45	100	400	4500(kg)
--------	----	-----	-----	----------

我们考虑木桶效应可以得到两辆货车的安全运输重量。考虑实际 2100 头的运输需求，按照每一类均取对应区间的最大重量可以计算得到，需要运输的总重量为：

$$M_{all} = 683 \times 20 + 1417 \times 108 = 166696 \text{ (kg)} \quad (26)$$

但是卡车每周的最大运输重量为：

$$M_{TrMax} = (9000 + 4500) \times 3 = 40500 \text{ (kg)} \quad (27)$$

从初步计算结果可以看出，如果在“旺季”的第一周周末直接将所有猪的存栏数从上限降到下限，需要运输能力与最大运输总量差距很大，所以确定这样的售卖方式是明显是不合理的。但是如果不结合实际情况去限定旺季每天销售副产品的数量，则会引入更多不确定性，所以我们转变思考方式。

在之前的分析中我们知道，在“旺季”一开始，猪的存栏量是达到存栏上限 11266 头的，考虑实际情况并为应对意外留出更多裕度，所以我们在题目给定的存栏下限 9000 基础上，增加了一部分最终将下限设定为 9166 头，所以在“旺季”对应的 27 周内，我们最终要争取将高于下限的 2100 头猪售卖完即可，按照“进 1 法”取平均后，这 2100 头需要在一周内销售的猪的数量为：

$$\left\lceil \frac{2100}{27} \right\rceil = [77.778] = 78 \text{ (头)} \quad (28)$$

在问题一分析计算过程中我们知道，每天出生的哺乳仔猪数量为 60 头，所以一周出生的仔猪数量为 420 头，此时为了保证货车的最大运力可以满足我们给定的销售计划，所以考虑极端情况，假设每周需要消耗的这些猪数量均由副产品的销售的消耗，但实际情况下，猪数量的消耗还有种猪的销售也会使其数量下降，但安全验算期间在这种极端情况下，每周需要售卖的猪数量为：

$$N_{week} = 420 + 78 = 498 \text{ (头)} \quad (29)$$

将 498 头按照三类猪的实际数量比来分配，由于暂时没有运行副产品的状态更新函数，所以假定三类猪的数量比保持最大存栏量的数量比，可以获得两类猪分别需要售卖的数量：

$$\begin{cases} \text{保育猪: } \left\lceil 498 \times \frac{2675}{8322} \right\rceil = [160.076] = 161 \text{ (头)} \\ \text{育肥猪: } 498 - 161 = 337 \text{ (头)} \end{cases} \quad (30)$$

还是按照每种类别下最大的体重来检验一周需要运输的重量：

$$M_{max} = 161 \times 20 + 337 \times 108 = 39616 \text{ (kg)} \quad (31)$$

可以看出，我们取两辆货车最低的运力，取需要运输的保育猪与育肥猪的最大重量时，货车完全可以满足对这两类副产品的运输，所以在后续制定任意副产品销售计划都可以符合货车运输的需求。

由于每周的消耗量 498 头并不一定完全需要通过副产品的销售去消耗，其中会有很多种猪也在被销售，所以修正后的每周销售副产品的总数为：

$$N_{week} = 420 + 78 - x \quad (32)$$

式中 x 代表每周实时得到的种猪的销售量。在得到每周需要销售的副产品总量的基础上，需要明确三类猪中每类猪各自的销售数量。采用的方法还是将每天需要销售的总数按照三种类别猪在每个周末现有数量的比例平均到各类猪去销售。假设在某周末保育猪以及育肥猪的数量分别为： P_1, P_2 ，此时各自需要销售的数量 S 如下：

$$\begin{cases} S_1 = N_{week} \times \frac{P_1}{P_1 + P_2} \\ S_2 = N_{week} - S_1 \end{cases} \quad (33)$$

在得到每周保育猪与育肥猪分别需要销售的副产品数量后，最后一步就是要在各自类别中合理选择应该销售哪些猪，选择的思路流程如图 3.20 所示。

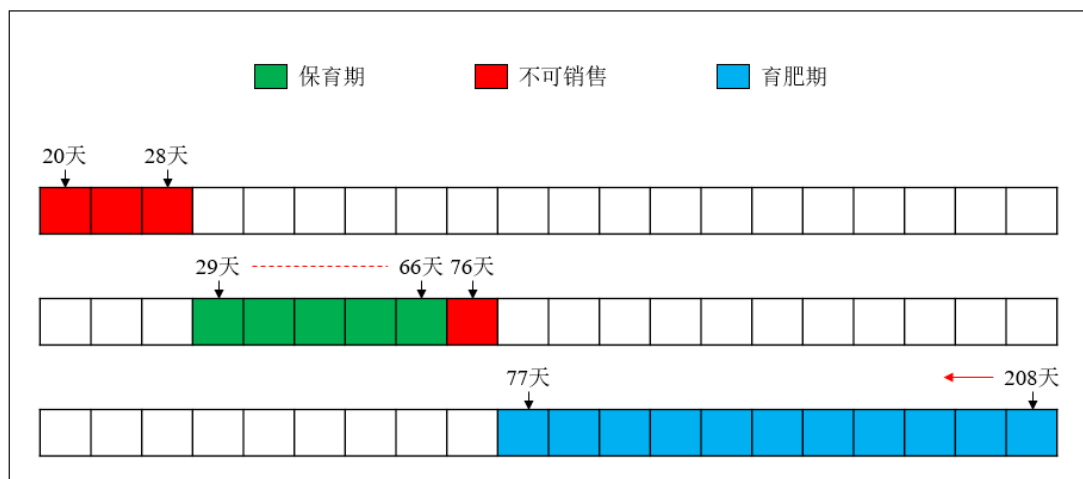


图 3.20 旺季阶段销售副产品的选定方法

我们采取的思路是，保育期的猪由于无法售卖 20~25kg 的猪，同时存在保育到育肥之间的过度，所以我们还是需要采用平均取样的销售方式以防止“截断”现象出现，直到数量达到本周需要出售的数量为止；育肥期的猪同样，由于之前的“淡季”已经买完了所有日龄大于 208 天的猪，所以售卖育肥猪时，应从 208 天开始往回选择，直到选择的数量达到本周需要出售的数量则停止选择。随着时间的逐步推进，当旺季的最后一周结束时便得到了所有旺季的副产品销售计划。

综上，通过制定两段“淡季”与一段“旺季”的副产品销售计划，便得到了全年 52 周的销售计划。我们将最终的得到的每周副产品销售计划放置于附件中，并命名为“问题二副产品每周销售计划.xls”。

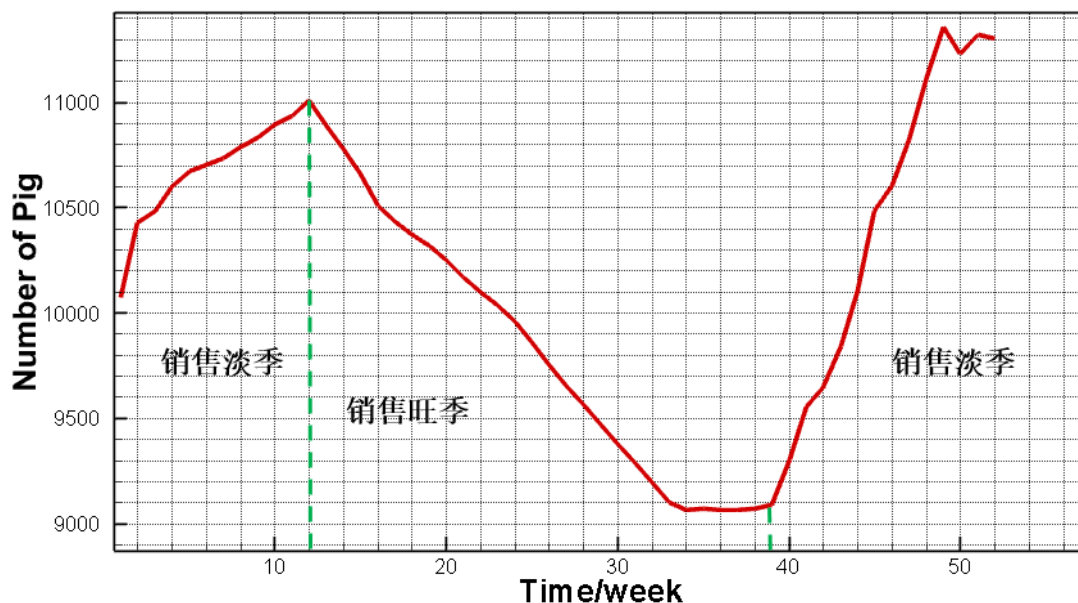


图 3.21 每周猪栏每周存猪数量统计

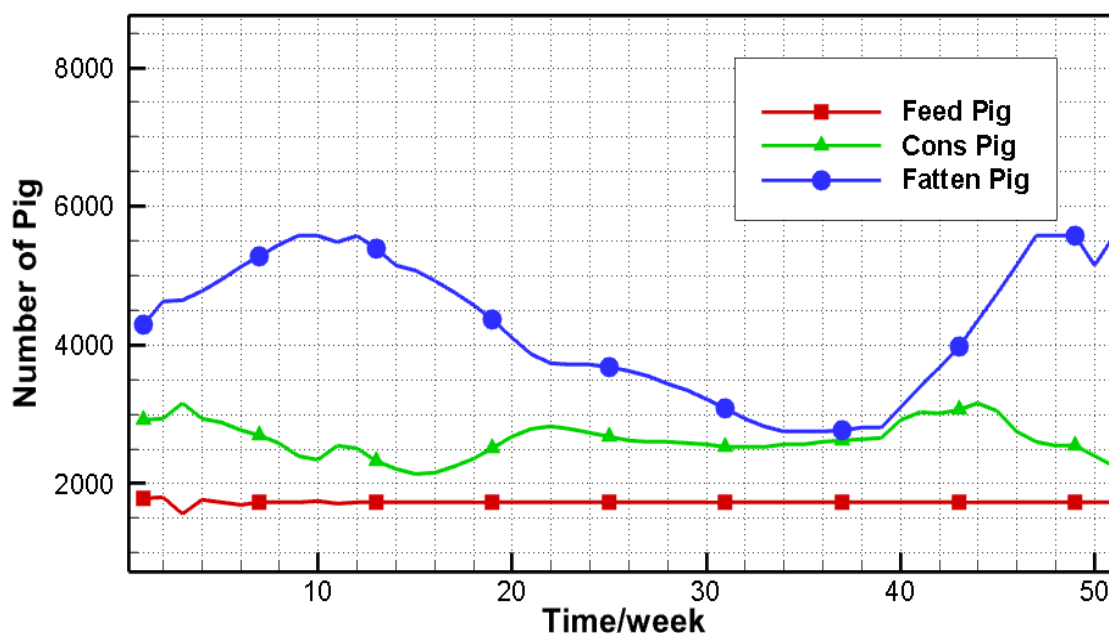
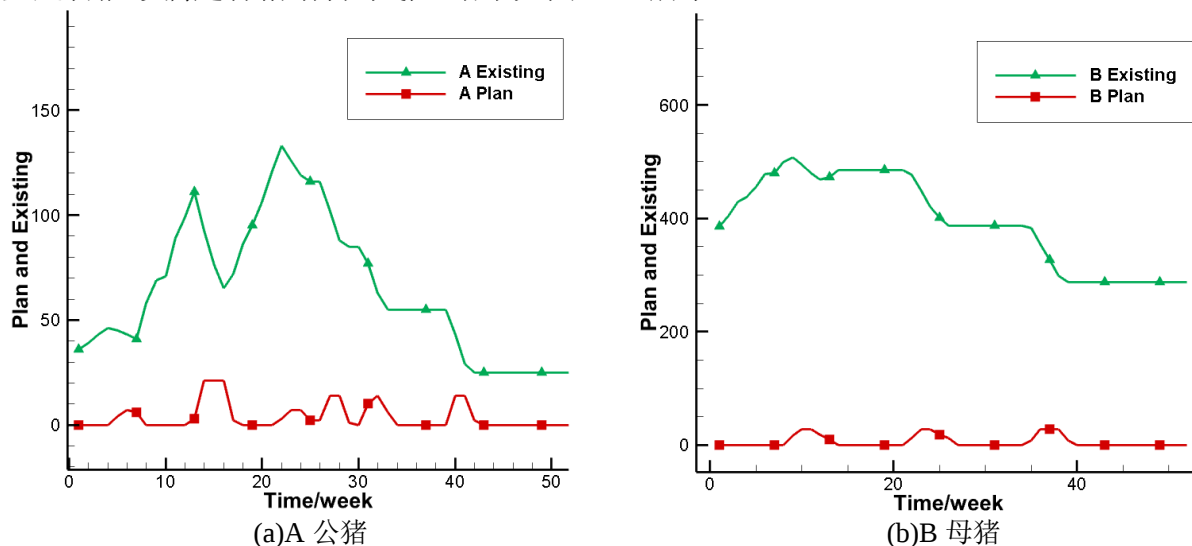


图 3.22 哺乳仔猪、保育猪与育肥猪每周存栏数量统计

从图 3.21 与 3.22 可以看出制定好销售计划后，我们执行状态更新函数可以得到每周现存猪的数量明显分为三个部分，在两段销售“淡季”内，现存数量每周都在上升，表现出囤积的现象，但都保持在最大存量上限以内；在销售“旺季”现存数量始终保持下降，最终在销售“旺季”结束前下降到最低存栏下限。这个变化趋势是符合我们建模思路的。

由于只有在问题二考虑副产品销售的基础上才能得到最终的状态更新函数，在此基础上，我们除了可以判断副产品的销售状况外，同样可以执行状态更新函数得到 A、C、E 系列公猪与 B、D、F 系母猪的现存数量，进而可以判断这些待售种猪随时间的存栏量是否能够满足种猪销售计划，结果如图 3.23 所示：



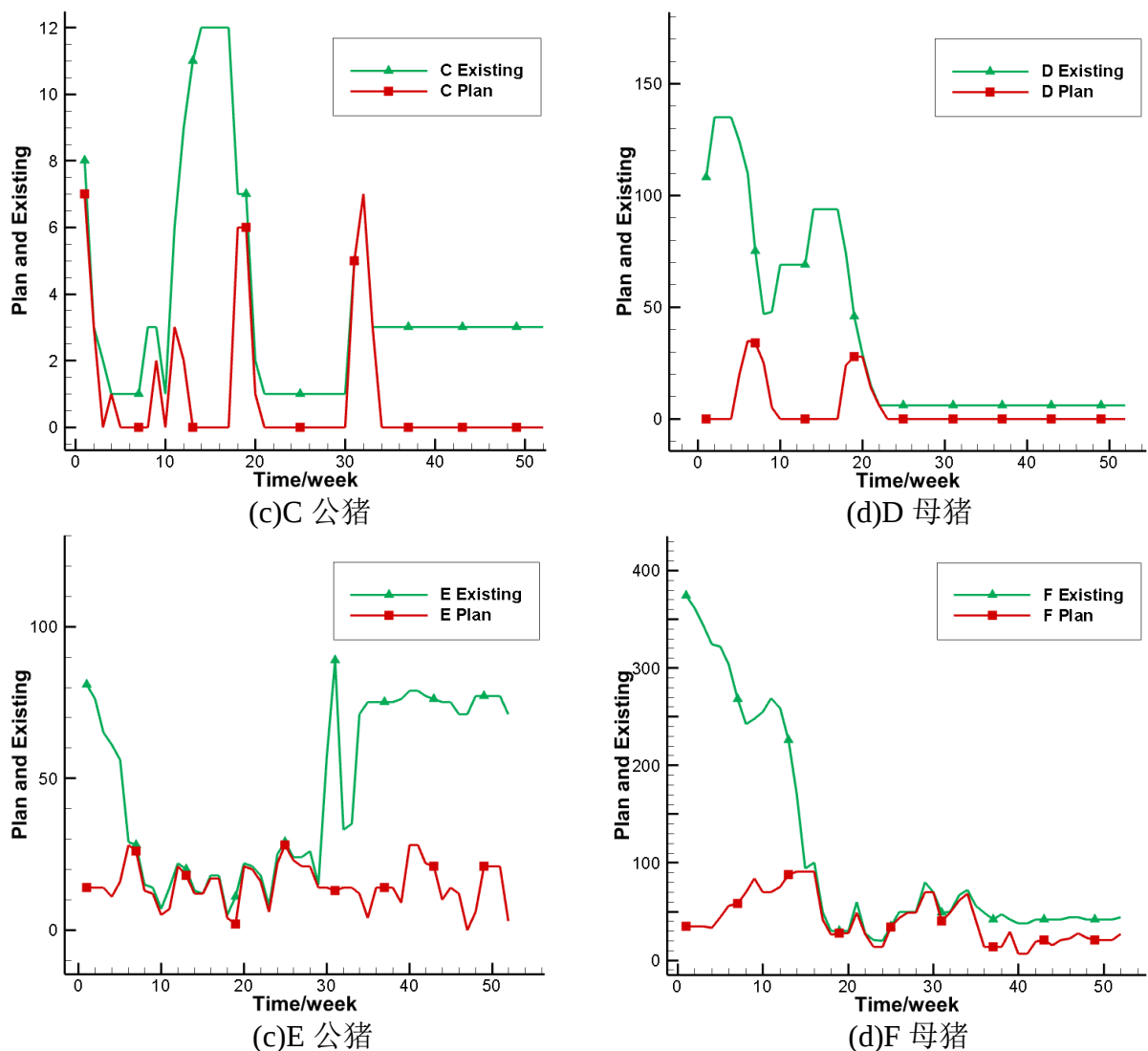


图 3.23 不同种类种猪现存数量与销售计划对比图

如图 3.23 所示，使用最终的状态更新函数更新后，每周的待售种猪现存量与种猪销售计划相比，可以看出每周待售种猪的现存量都可以满足种猪销售计划要求。

3.2.3 模型检验

问题二建立的模型需要检验以下两部分内容：

1. 淡季销售的过程中，是否会出现某天猪的现有存栏数低于最低下限或高于各类最高上限的情况？

由于我们建立模型的出发点，在考虑副产品销售的过程中，每次都对存栏这个约束进行了检验，在淡季由于我们每周卖出的较少，所以实时检查了是否会出现高于最大存栏量的情况，将下一周高于存栏量的部分全部销售出去；在旺季时，我们每次出售的副产品较多，但在销售时，我们每天执行状态更新函数，确保其不会低于存栏下限约束。

2. 旺季制定的销售计划是否能满足最大运力需求

在模型建立与求解部分已经经过检验，我们选择的售卖方式是在周末当天销售，考虑两辆卡车的运输能力时，我们按照附件 5 中给出的卡车运输能力下限来换算出了卡车的运输重量，按照下限换算出的重量为 40500kg，但实际的最大运输重量肯定是高于 40500 千克的。同时我们计算出每周需要运输猪的极限数量为 498 头，并按照每周每种猪售卖数量，并取每个区间重量的最大值换算出一周需要被运输的极限重量为 39616kg，并且每周需要售卖的猪的数量并不是全都要副产品去填充，其中还有一些种猪是要按照

种猪销售计划被出售的，所以我们判定货车车的运输能力完全可以满足我们在销售“旺季”设置的副产品销售计划。

综上，经问题二建立的销售模型，可以使得货车满足我们制定的销售计划，具备一定的合理性与有效性。

3.3 问题三模型建立与求解

问题三的建模思路如图 3.24 所示。

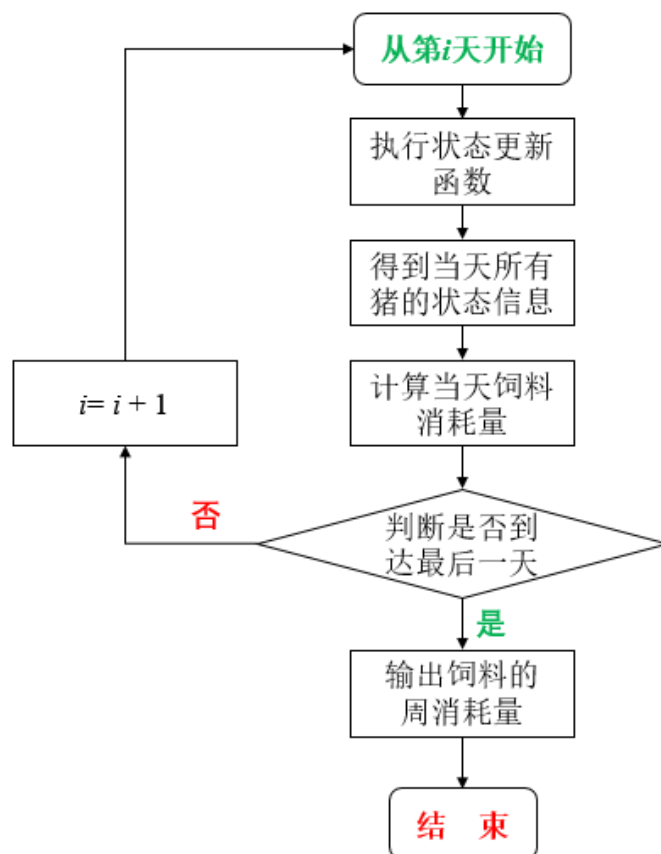


图 3.24 问题三模型建立与求解流程图

在问题一与问题二中，我们针对不同的问题分别给出了不同的状态更新函数。所以在计算第三问饲料消耗之前，我们已经知道了每天猪的状态信息，包括每天处于不同类别猪的个数、日龄以及体重等信息。由于附件 1 已经给出了猪的饲喂程序，通过对附件 1 的理解，其给出了处于不同日龄的每头猪每天要消耗对应饲料的重量。

有三种饲料相对特殊，分别为饲料 1-Sow、饲料 2-Sow 以及饲料 2-Boar。其中饲料 1-Sow 与饲料 2-Sow 是生产母猪吃的，而饲料 2-Boar 是给生产公猪吃的，所以问题三我们需要做的就是得到每天状态的基础上，将处于不同类别，不同日龄的猪要消耗的饲料量相加即可，最后按天累计便可得到每周的饲料消耗量。

具体对应的状态划分如下：

表 3.11 不同状态的猪对应的饲料及每天的饲料消耗量

生产猪 非生产猪

日龄 /天	饲料 1-Sow	饲料 2-Sow	饲料 3	饲料 4	饲料 5	饲料 6	饲料 7	饲料 8	饲料 2-Boar
0~20									
21~28			0.3kg						
29~45				0.6kg					

46~70					1.1kg				
71~100						1.6kg			
101~130							2.25kg		
130~168								2.4kg	
>168								2.4kg	
生产母猪 1	2.6kg								
生产母猪 2		5.12kg							
生产公猪									2.5kg

最终将得到的饲料消耗结果放于附件中，其文件命名为“问题三饲料消耗结果”。

4 结论

本文共解决了三个问题，问题一是制定配种计划以及生产公猪母猪的更新计划。我们将这个问题分为三个部分来依次处理，首先是考虑题目附件中给出的 2019 年种猪的月销售计划，结合实际，我们将销售计划按照均值分配的方法转换为每天的销售计划，种猪的售卖采用按天销售的模式，在得到每天销售计划的基础上，给定猪的状态更新模型，通过状态更新模型可以实时得到每天所有猪的状态信息，包括日龄、体重、所处的状态以及不同状态之间的转换关系。随着状态更新函数的时间推进，可以检查当天的待售种猪数量是否满足当天的销售计划，如果不满足则通过时间反推得到之前的配种计划。通过这一步得到的是考虑种猪销售计划的“初步配种方案”；其次要考虑生产公猪与生产母猪的更新问题，按照题目的更新要求，更新了所有生产公猪以及 65% 的日龄大于 952 天的生产母猪(配种后工作两年的生产母猪)。由于生产母猪的数量庞大，但是初始后备生产母猪仅有 54 头，所以要在“初步配种方案”的基础上继续增加配种，来满足对生产母猪的更新。经过这一步得到的配种方案我们将其称为“初步修正的配种方案”；最后在满足种猪销售计划与生产猪更新的基础上，为了给后续副产品的销售留出更大的裕度，我们选择继续扩充产能，得到“最终配种方案”。经过验算，我们扩充的目标是每天保证有 3 窝哺乳仔猪出生，并且尽可能使 A 公猪与 B 母猪多配，这样产出的 F 类哺乳仔猪个数最多。以保证在问题二销售副产品时尽可能多地获利。

问题二要在问题一得到“最终配种方案”的基础上考虑副产品的销售问题，此时我们需要在考虑添加副产品销售的基础上，来修正问题一的状态更新函数。确定具体的副产品销售方案之前，分析附件 4 给出的“副产品售价预测”，我们将 2019 年副产品的销售方案分为“淡季”与“旺季”两个时期分别考虑。在“淡季”时我们的目标是保证现有存栏猪数量不超过最大存栏上限，并且首先按照模型制定的平均销售方案来出售保育猪，以及育肥猪中日龄超过 208 天的猪。这样做的目的是在猪的数量一定的情况下，尽量保证在“旺季”销售更多的猪以实现利润的最大化；在“旺季”我们的目标是尽量多卖，但要保证每天的存栏的猪总量不能低于存栏下限并且满足两辆货车的最大运力约束。在确定“旺季”具体销售方案时，模型采取的方法是首先采用平均销售原则来出售保育猪，其次出售日龄较大的育肥猪。“淡季”与“旺季”两种不同时间区间的销售方案汇总后最终得到了 2019 年内 52 周的副产品销售计划。

问题三在问题二得到的修正状态更新函数的基础上，首先得到了每天的所有猪的状态，包括猪的日龄、类别以及不同类别间的转换关系。获得这些状态量后，结合附件 1 的“饲喂程序”找到不同日龄、状态的猪对应的饲料类别，从而可计算出每天的饲料总消耗，通过时间累积最终得到了每周的饲料消耗状态。

综上，本文针对问题建立的三个模型制定了符合种猪销售计划、满足生产猪更新方案以及为副产品销售留出足够裕度的配种计划，在此基础上考虑实际情况，制定了合理的副产品销售计划，经模型检验满足货车最大运力约束，并最终计算出 2019 年每周饲料消耗量。以上证明本文所建模型的合理性与有效性。

5 模型优点与不足

5.1 模型优点

1. 模型一在制定 2019 年配种计划的过程中，考虑到了 2020 年的种猪销售情况。由于我们实现了 2019 年全年配种，所以为下一年种猪的销售留出了足够的裕度；
2. 模型一的整个思路过程是相对复杂的，我们将其合理地拆解为三个部分依次考虑，有效降低了模型以及程序设计的复杂度，并且每个阶段可以分别对模型进行独立检验，程序设计结构化更加明显；
3. 在考虑模型一的状态更新时，对一部分生产母猪的更新使用我们自己定义的“自留母猪”来补充，这样与待售种猪区分开，有效降低了后续计算分析的复杂度，并且我们在更新完生产猪的基础上还进一步扩充了 54 头后备母猪，为下一年的生产母猪更新留出了一定裕度；
4. 模型一在扩充产能时，经过验算提出了一种更加简洁的扩充方式，首先可以保证扩充后不超过存栏上限，其次扩充方式更加简洁；
5. 模型二在制定副产品销售方案之前首先考虑副产品销售，改进了模型一的状态更新函数，进而可以更新每天存栏猪的数量、状态等参数，更加便于后续销售模型的建立；
6. 制定副产品销售计划时在考虑 2019 年预测的每个月市场价格的基础上，我们将销售过程分为“淡季”和“旺季”两个不同的时间区间来考虑，并且在不同的时间区间内使用不同的销售方式来保证利润的最大化。

5.2 模型缺点

1. 在制定配种计划与销售方案的过程中为了避免更为复杂的情况，只是固定地乘了分娩率、死亡率以及选种率与自留率，而没有考虑其中存在的随机性；
2. 在制定 2019 年全年配种计划的过程中，由于题目中没有给定 2020 年的销售计划，所以我们引入假设 2020 年与 2019 年的销售计划相同。这样的处理方式可能会引入部分误差；
3. 在考虑猪的存栏上限时，我们只是给定了上限猪的数量而没有根据猪所占的体积去换算，这个过程引入了部分误差。

参考文献

- [1] 周维, 徐小娟, 王寿昆. 母猪群的管理方案[J]. 今日养猪业, 2008(3):47-48.
- [2] 佚名. 100 头基础母猪猪场设计方案[J]. 农业知识, 2004(36):23-23.
- [3] 周明君, 张海筠, 王爱清. 种猪营销方案[J]. 湖北畜牧兽医, 2002(5):15-15.
- [4] A.G.deVries, 王余平. 现代猪优化育种方案(上)[J]. 中国畜牧兽医, 1991(6):5-7.
- [5] 王宏, 张秉慧. 后备母猪的营养配置方案[J]. 国外畜牧学-猪与禽, 2001(1):15-17.
- [6] 王升鉴. 猪副产品的销售价格应该放开[J]. 价格月刊, 1984(8):29-31.

附录

主程序

```
clc;
tic
Origin_data_to_num;
day_all=364;%输入不能超过364
day_num=1;
Birth_plan_for_sale=zeros(day_all,7);
for i=1:1:day_all
    Birth_plan_for_sale(i,1)=i;
end
saleplan_all=zeros(52,4);
Birth_plan_all=produce33;    %输入生产计划表
plan_of_sale=Sale_plan;    %输入销售计划表
birth_plan_for_sale_2=Birth_plan_for_sale_single_last;    %输入为满足销售计划的
生产计划表
flag=1;
Now_data=Chushihua(Origin_num_data);
numberpig=[];
costfood=[];
connum=[];
number_for_sale=zeros(52,6);
for i=1:1:52
    for j=(i-1)*7+1:1:i*7
        A_birth=Birth_plan_all(j,1:6);
        B_sale_zhong_plan=plan_of_sale(j,2:7);
        C_birth_plan_for_sale=birth_plan_for_sale_2(j,2:7);

[Now_data,flag2,num_for_add,num22]=Update_Last(Now_data,A_birth,B_sale_zhong_plan,
C_birth_plan_for_sale,j);
        Now_data=day_by_day(Now_data,day_num);
    end
    number_for_sale(i,:)=num22;
    Now_data2=Now_data;%得到当前周末的状态
    if i<52
        for k=i*7+1:1:(i+1)*7
            A_birth=Birth_plan_all(k,1:6);
            B_sale_zhong_plan=plan_of_sale(k,2:7);
            C_birth_plan_for_sale=birth_plan_for_sale_2(k,2:7);
[Now_data2,flag2,num_for_add]=Update_Last(Now_data2,A_birth,B_sale_zhong_plan,
C_birth_plan_for_sale,k);
            Now_data2=day_by_day(Now_data2,day_num);%得到下一周末的状态
        end
    end

    if i<=12
[Now_data,saleplan]=poorseason(Now_data,Now_data2);
```

```

saleplan_all(i,:)=saleplan;
size(Now_data,1)
numberpig=[numberpig;size(Now_data,1)];
cost=costcal(Now_data);
costfood=[costfood;cost];
small=Now_data(find(Now_data(:,5)==3),:);%初始猪的状态
nsmall=size(small,1);
nurse=Now_data(find(Now_data(:,5)==4),:);
nnurse=size(nurse,1);
fat=[Now_data(find(Now_data(:,5)==6),:);Now_data(find(Now_data(:,5)==5),:)];
nfat=size(fat,1);
nsmall=ceil(nsmall*1.2);
nnurse=ceil(nnurse*1.2);
nfat=ceil(nfat*1.2);
delta=[nsmall,nnurse,nfat,nsmall+nnurse,nsmall+nnurse+nfat];
connum=[connum;delta];
end
if i<52 && i>39
[Now_data,saleplan]=poorseason(Now_data,Now_data2);
saleplan_all(i,:)=saleplan;
size(Now_data,1)
numberpig=[numberpig;size(Now_data,1)];
cost=costcal(Now_data);
costfood=[costfood;cost];
nsmall=size(small,1);
nurse=Now_data(find(Now_data(:,5)==4),:);
nnurse=size(nurse,1);
fat=[Now_data(find(Now_data(:,5)==6),:);Now_data(find(Now_data(:,5)==5),:)];
nfat=size(fat,1);
nsmall=ceil(nsmall*1.2);
nnurse=ceil(nnurse*1.2);
nfat=ceil(nfat*1.2);
delta=[nsmall,nnurse,nfat,nsmall+nnurse,nsmall+nnurse+nfat];
connum=[connum;delta];
end
if i>12&&i<40
[Now_data,saleplan]=richseason(Now_data);
saleplan_all(i,:)=saleplan;
size(Now_data,1)
numberpig=[numberpig;size(Now_data,1)];
cost=costcal(Now_data);
costfood=[costfood;cost];
nsmall=size(small,1);
nurse=Now_data(find(Now_data(:,5)==4),:);
nnurse=size(nurse,1);
fat=[Now_data(find(Now_data(:,5)==6),:);Now_data(find(Now_data(:,5)==5),:)];
nfat=size(fat,1);
nsmall=ceil(nsmall*1.2);
nnurse=ceil(nnurse*1.2);
nfat=ceil(nfat*1.2);

```

```

delta=[nsmall,nnurse,nfat,nsmall+nnurse,nsmall+nnurse+nfat];
connum=[connum;delta];
    end
%     if i==52
%         111111
%     end
    if i>51
        [Now_data,saleplan]=richseason(Now_data);
        saleplan_all(i,:)=saleplan;
        size(Now_data,1)
        numberpig=[numberpig;size(Now_data,1)];
        cost=costcal(Now_data);
        costfood=[costfood;cost];
        nsmall=size(small,1);
        nurse=Now_data(find(Now_data(:,5)==4),:);
        nnurse=size(nurse,1);
        fat=[Now_data(find(Now_data(:,5)==6),:);Now_data(find(Now_data(:,5)==5),:)];
        nfat=size(fat,1);
        nsmall=ceil(nsmall*1.2);
        nnurse=ceil(nnurse*1.2);
        nfat=ceil(nfat*1.2);
        delta=[nsmall,nnurse,nfat,nsmall+nnurse,nsmall+nnurse+nfat];
        connum=[connum;delta];
    end

end
tocc

```

初始化函数

```

function  Origin_num_data_updata=Chushihua( Origin_num_data_updata)
[m,~]=size(Origin_num_data_updata);
%%%设定已经满足条件的可卖种猪的更新
for i=1:1:m
    if
        abs(Origin_num_data_updata(i,6)-1)<0.001                                &&
abs(Origin_num_data_updata(i,5)-6)<0.001 && abs(Origin_num_data_updata(i,4)-0)<0.001
        if Origin_num_data_updata(i,3)>15*7
            Origin_num_data_updata(i,5)=7;
        end
    end

    if
        abs(Origin_num_data_updata(i,6)-1)<0.001                                &&
abs(Origin_num_data_updata(i,5)-6)<0.001 && abs(Origin_num_data_updata(i,4)-1)<0.001
        if Origin_num_data_updata(i,3)>23*7
            Origin_num_data_updata(i,5)=7;
        end
    end
end
end
%%% 对公猪进行标记

Origin_num_data_updata=Boar_Replace_sign(Origin_num_data_updata);

```

end

猪群状态更新函数

```
function[Now_data,flag2,num_for_add,number_for_sale]=Update_Last>Last_day_data,  
A_chusheng,B_sale_zhong_plan,C_birth_plan_for_sale22,day)
```

```
A1=A_chusheng(1,1);    %出生的 A 母猪/公猪个数
```

```
B1=A_chusheng(1,2);    %出生的 B 母猪/公猪个数
```

```
C1=A_chusheng(1,3);    %出生的 C 母猪/公猪个数
```

```
D1=A_chusheng(1,4);    %出生的 D 母猪/公猪个数
```

```
E1=A_chusheng(1,5);    %出生的 E 母猪/公猪个数
```

```
F1=A_chusheng(1,6);    %出生的 F 母猪/公猪个数
```

```
A2_boar=B_sale_zhong_plan(1,1); %要卖出的 A 公种猪
```

```
B2_sow=B_sale_zhong_plan(1,2);   %要卖出的 B 母种猪
```

```
C2_boar=B_sale_zhong_plan(1,3); %要卖出的 C 公种猪
```

```
D2_sow=B_sale_zhong_plan(1,4);   %要卖出的 D 母种猪
```

```
E2_boar=B_sale_zhong_plan(1,5); %要卖出的 E 公种猪
```

```
F2_sow=B_sale_zhong_plan(1,6);   %要卖出的 F 母种猪
```

```
A3_boar=C_birth_plan_for_sale22(1,1); %要出生的用于种猪的 A 公猪的个数
```

```
B3_sow=C_birth_plan_for_sale22(1,2);   %要出生的用于种猪的 B 母猪的个数
```

```
C3_boar=C_birth_plan_for_sale22(1,3); %要出生的用于种猪的 C 公猪的个数
```

```
D3_sow=C_birth_plan_for_sale22(1,4);   %要出生的用于种猪的 D 母猪的个数
```

```
E3_boar=C_birth_plan_for_sale22(1,5); %要出生的用于种猪的 E 公猪的个数
```

```
F3_sow=C_birth_plan_for_sale22(1,6);   %要出生的用于种猪的 F 母猪的个数
```

%% 新增小猪猪

```
Last_day_data=child_add_last( Last_day_data,1,A1,A3_boar,1,day);           %生下 A  
纯种小猪猪;
```

```
[m,~]=size>Last_day_data);
```

```
for i=1:1:m
```

```
    if Last_day_data(i,7)>0
```

```
        Last_day_data(i,7)=1;
```

```
    end
```

```
end
```

```
Last_day_data=child_add_last( Last_day_data,2,B1,B3_sow,0,day);           %生下 B  
纯种小猪猪;
```

```
Last_day_data=child_add_last( Last_day_data,3,C1,C3_boar,1,day);           %生下 C  
纯种小猪猪;
```

```
Last_day_data=child_add_last( Last_day_data,4,D1,D3_sow,0,day);           %生下 D  
纯种小猪猪;
```

```

        Last_day_data=child_add_last( Last_day_data,5,E1,E3_boar,1,day);           %生下 E
小猪猪
        Last_day_data=child_add_last( Last_day_data,6,F1,F3_sow,0,day);           %生下 F 小
猪猪

%% 进行母猪种猪和公猪种猪的选种
%进行母猪的选种，选中率大约 65%,选种时间为 15 周
    Last_day_data=Select_sow_last(Last_day_data,1,0,15*7,0.65);%对 A 母猪进行选种，
选中率为 65%，剩下的全放到副产品里面
    Last_day_data=Select_sow_last(Last_day_data,2,0,15*7,0.65);%对 B 母猪进行选种，
同上
    Last_day_data=Select_sow_last(Last_day_data,3,0,15*7,0.65);%对 C 母猪进行选种，
同上
    Last_day_data=Select_sow_last(Last_day_data,4,0,15*7,0.65);%对 D 母猪进行选种，
同上
    Last_day_data=Select_sow_last(Last_day_data,6,0,15*7,0.65);%对 F 母猪进行选种，同
上
    %进行公猪的选种，选中率大约 40%,选种时间为 23 周
    Last_day_data=Select_boar_last(Last_day_data,5,1,23*7,0.40);%对 E 公猪进行选种，
选中率为 40%，剩下的全放到副产品里面
    Last_day_data=Select_boar_last(Last_day_data,1,1,23*7,0.40);%对 A 公猪进行选种，
选中率为 40%，剩下的全放到副产品里面
    Last_day_data=Select_boar_last(Last_day_data,2,1,23*7,0.40);%对 B 公猪进行选种，
同上
    Last_day_data=Select_boar_last(Last_day_data,3,1,23*7,0.40);%对 C 公猪进行选种，
同上
    Last_day_data=Select_boar_last(Last_day_data,4,1,23*7,0.40);%对 D 公猪进行选种，
同上
%% 进行生产公猪的更新
    Last_day_data=Replace_boar(Last_day_data,1); %对 A 种类生产公猪进行更新
    Last_day_data=Replace_boar(Last_day_data,2); %对 B 种类生产公猪进行更新
    Last_day_data=Replace_boar(Last_day_data,3); %对 C 种类生产公猪进行更新
    Last_day_data=Replace_boar(Last_day_data,4); %对 D 种类生产公猪进行更新
%% 进行生产母猪的更新
    day_update_sow=day;
    Last_day_data=Replace_sow(Last_day_data,1,day_update_sow); %对 A 种类生产母
猪进行更新
    Last_day_data=Replace_sow(Last_day_data,2,day_update_sow); %对 B 种类生产母
猪进行更新
    Last_day_data=Replace_sow(Last_day_data,3,day_update_sow); %对 C 种类生产母
猪进行更新
    Last_day_data=Replace_sow(Last_day_data,4,day_update_sow); %对 D 种类生产母
猪进行更新
    % 进行后备母猪的更新
    day_update_sow_pre=day;

```

Last_day_data=Replace_sow_pre>Last_day_data,1,day_update_sow_pre); %对 A 种类
后备母猪进行更新

Last_day_data=Replace_sow_pre>Last_day_data,2,day_update_sow_pre); %对 B 种类
后备母猪进行更新

Last_day_data=Replace_sow_pre>Last_day_data,3,day_update_sow_pre); %对 C 种类
后备母猪进行更新

Last_day_data=Replace_sow_pre>Last_day_data,4,day_update_sow_pre); %对 D 种类
后备母猪进行更新

%% 进行保育期小猪的更新

Last_day_data=nurse_update>Last_day_data,1,29); %对 A 种类保育期小猪进行更新

Last_day_data=nurse_update>Last_day_data,2,29); %对 B 种类保育期小猪进行更新

Last_day_data=nurse_update>Last_day_data,3,29); %对 C 种类保育期小猪进行更新

Last_day_data=nurse_update>Last_day_data,4,29); %对 D 种类保育期小猪进行更新

Last_day_data=nurse_update>Last_day_data,5,29); %对 E 种类保育期小猪进行更新

Last_day_data=nurse_update>Last_day_data,6,29); %对 F 种类保育期小猪进行更新

%% 进行育肥期/待售种猪小猪的更新

Last_day_data=finish_update>Last_day_data,1,77); %对 A 种类育肥期/待售种猪小猪
进行更新

Last_day_data=finish_update>Last_day_data,2,77); %对 B 种类育肥期/待售种猪小猪
进行更新

Last_day_data=finish_update>Last_day_data,3,77); %对 C 种类育肥期/待售种猪小猪
进行更新

Last_day_data=finish_update>Last_day_data,4,77); %对 D 种类育肥期/待售种猪小猪
进行更新

Last_day_data=finish_update>Last_day_data,5,77); %对 E 种类育肥期/待售种猪小猪
进行更新

Last_day_data=finish_update>Last_day_data,6,77); %对 F 种类育肥期/待售种猪小猪
进行更新

%% 进行副产品哺乳仔猪个数的统计

num_for_child=number_for_child_last>Last_day_data);

num_A_sow_for_child=num_for_child(1,1);

num_A_boar_for_child=num_for_child(1,2);

num_B_sow_for_child=num_for_child(2,1);

num_B_boar_for_child=num_for_child(2,2);

num_C_sow_for_child=num_for_child(3,1);

num_C_boar_for_child=num_for_child(3,2);

num_D_sow_for_child=num_for_child(4,1);

num_D_boar_for_child=num_for_child(4,2);

num_E_sow_for_child=num_for_child(5,1);

num_E_boar_for_child=num_for_child(5,2);

num_F_sow_for_child=num_for_child(6,1);

num_F_boar_for_child=num_for_child(6,2);

num_for_child_all=num_for_child(7,2);

%% 进行副产品保育期小猪个数的统计

num_for_nurse=number_for_nurse_last>Last_day_data);

num_A_sow_for_nurse=num_for_nurse(1,1);

```

num_A_boar_for_nurse=num_for_nurse(1,2);
num_B_sow_for_nurse=num_for_nurse(2,1);
num_B_boar_for_nurse=num_for_nurse(2,2);
num_C_sow_for_nurse=num_for_nurse(3,1);
num_C_boar_for_nurse=num_for_nurse(3,2);
num_D_sow_for_nurse=num_for_nurse(4,1);
num_D_boar_for_nurse=num_for_nurse(4,2);
num_E_sow_for_nurse=num_for_nurse(5,1);
num_E_boar_for_nurse=num_for_nurse(5,2);
num_F_sow_for_nurse=num_for_nurse(6,1);
num_F_boar_for_nurse=num_for_nurse(6,2);
num_for_nurse_all=num_for_nurse(7,2);
%% 进行副产品育肥期猪个数的统计
num_for_finish=number_for_finish_last(Last_day_data);
num_A_sow_for_finish=num_for_finish(1,1);
num_A_boar_for_finish=num_for_finish(1,2);
num_B_sow_for_finish=num_for_finish(2,1);
num_B_boar_for_finish=num_for_finish(2,2);
num_C_sow_for_finish=num_for_finish(3,1);
num_C_boar_for_finish=num_for_finish(3,2);
num_D_sow_for_finish=num_for_finish(4,1);
num_D_boar_for_finish=num_for_finish(4,2);
num_E_sow_for_finish=num_for_finish(5,1);
num_E_boar_for_finish=num_for_finish(5,2);
num_F_sow_for_finish=num_for_finish(6,1);
num_F_boar_for_finish=num_for_finish(6,2);
num_for_finish_all=num_for_finish(7,2);
%% 进行可卖种猪个数的统计
num_for_sale_real=number_for_sale_real(Last_day_data);
num_A_sow_for_sale_real=num_for_sale_real(1,1);
num_A_boar_for_sale_real=num_for_sale_real(1,2);
num_B_sow_for_sale_real=num_for_sale_real(2,1);
num_B_boar_for_sale_real=num_for_sale_real(2,2);
num_C_sow_for_sale_real=num_for_sale_real(3,1);
num_C_boar_for_sale_real=num_for_sale_real(3,2);
num_D_sow_for_sale_real=num_for_sale_real(4,1);
num_D_boar_for_sale_real=num_for_sale_real(4,2);
num_E_sow_for_sale_real=num_for_sale_real(5,1);
num_E_boar_for_sale_real=num_for_sale_real(5,2);
num_F_sow_for_sale_real=num_for_sale_real(6,1);
num_F_boar_for_sale_real=num_for_sale_real(6,2);
num_for_sale_real_all=num_for_sale_real(7,2);
number_for_sale=zeros(1,6);
number_for_sale(1,1)=num_A_boar_for_sale_real;
number_for_sale(1,2)=num_B_sow_for_sale_real;
number_for_sale(1,3)=num_C_boar_for_sale_real;
number_for_sale(1,4)=num_D_sow_for_sale_real;
number_for_sale(1,5)=num_E_boar_for_sale_real;
number_for_sale(1,6)=num_F_sow_for_sale_real;
%% 进行种猪的售卖

```

```

flag2=0;
AAA=zeros(1,6);
num_for_add=zeros(1,6);
num_for_add(1,1)=num_A_boar_for_sale_real-A2_boar;
num_for_add(1,2)=num_B_sow_for_sale_real-B2_sow;
num_for_add(1,3)=num_C_boar_for_sale_real-C2_boar;
num_for_add(1,4)=num_D_sow_for_sale_real-D2_sow;
num_for_add(1,5)=num_E_boar_for_sale_real-E2_boar;
num_for_add(1,6)=num_F_sow_for_sale_real-F2_sow;

if A2_boar < num_A_boar_for_sale_real+1
    AAA(1,1)=1;

    if A2_boar > 0.1
        Last_day_data=sale_zhong(Last_day_data,1,1,A2_boar);
    end
end
if B2_sow < num_B_sow_for_sale_real+1
    AAA(1,2)=1;
    if B2_sow > 0.1
        Last_day_data=sale_zhong(Last_day_data,2,0,B2_sow);
    end
end
if C2_boar < num_C_boar_for_sale_real+1
    AAA(1,3)=1;
    if C2_boar > 0.1
        Last_day_data=sale_zhong(Last_day_data,3,1,C2_boar);
    end
end
if D2_sow < num_D_sow_for_sale_real+1
    AAA(1,4)=1;
    if D2_sow > 0.1
        Last_day_data=sale_zhong(Last_day_data,4,0,D2_sow);
    end
end
if E2_boar < num_E_boar_for_sale_real+1
    AAA(1,5)=1;
    if E2_boar > 0.1
        Last_day_data=sale_zhong(Last_day_data,5,1,E2_boar);
    end
end
if F2_sow < num_F_sow_for_sale_real+1
    AAA(1,6)=1;
    if F2_sow > 0.1
        Last_day_data=sale_zhong(Last_day_data,6,0,F2_sow);
    end
end
for j=1:1:6

```

```

        if AAA(1,j)<0.5
            flag2=1;
            break
        end
    end
    Now_data=Last_day_data;
End

```

更新新生小猪状态函数

```

function D=child_add_last(A,kind,num33,C3,sex,day)
[m,~]=size(A);
% if C3>5.5
%     if abs(day-5)<0.1
%         C3=C3+20;
%     end
% end
if day>59 && day<62
    C3=C3+1;
end
C3=round(C3);

num_sow=round(0.9*num33);
num_boar=num_sow;
B=zeros(m+num_sow,8);
B(1:m,:)=A;
if num_sow >0
    for i=m+1:1:m+num_sow
        B(i,1)=i;    B(i,2)=kind;    B(i,3)=1;    B(i,4)=0;    B(i,5)=3;    B(i,6)=2;
        B(i,7)=0;B(i,8)=0;

    end
    m2=size(B,1);
    if abs(sex-0)<0.001 && C3>0
        for i=1:1:C3

            B(m2-i+1,1)=m2-i+1;    B(m2-i+1,2)=kind;    B(m2-i+1,3)=1;    B(m2-i+1,4)=0;
            B(m2-i+1,5)=3;
            B(m2-i+1,6)=1;    B(m2-i+1,7)=1; B(m2-i+1,8)=0;
        end
    end
end
m2=size(B,1);
C=zeros(m2+num_boar,8);
C(1:m2,:)=B;
if num_boar >0
    for i=m2+1:1:m2+num_boar
        C(i,1)=i;    C(i,2)=kind;    C(i,3)=1; C(i,4)=1;    C(i,5)=3;    C(i,6)=2; C(i,7)=0;C(i,8)=0;
    end
    m3=size(C,1);

```

```

if abs(sex-1)<0.001 && C3>0
    for i=1:1:C3
        C(m3-i+1,1)=m3-i+1;    C(m3-i+1,2)=kind;    C(m3-i+1,3)=1;    C(m3-i+1,4)=1;
C(m3-i+1,5)=3;
        C(m3-i+1,6)=1;    C(m3-i+1,7)=1; C(m3-i+1,8)=0;
    end
end
end
D=C;
end

```

对生产公猪进行更新函数

```

function A=Replace_boar(A,kind)

[m,~]=size(A);
num_A_boar_for_replace=0;
for i=1:1:m
    if abs(A(i,2)-kind)<0.00001 && abs(A(i,5)-8)<0.00001
        num_A_boar_for_replace=num_A_boar_for_replace+1;
        A_boar_for_replace( num_A_boar_for_replace,:)=A(i,:);
    end
end
if num_A_boar_for_replace>0
    A_boar_for_replace=sortrows(A_boar_for_replace,3);
    num1=size(A_boar_for_replace,1);
    for i=1:1:num1
        if      abs(A(i,2)-kind)<0.00001      &&      abs(A(i,5)-1)<0.00001      &&
abs(A(i,3)-666)<0.00001&& abs(A(i,7)-0)<0.00001
            A(i,6)=2;
            A(i,5)=5;
            aa=A_boar_for_replace(num1,1);
            A(aa,6)=1;
            A(aa,5)=1;
        end
    end
end
end
end

```

对生产母猪进行更新函数

```

function A=Replace_sow(A,kind,day)
kind1=kind+1;
Birth_plan_for_sow_update1=zeros(700,7);
m2=size(Birth_plan_for_sow_update1,1);
for i=1:1:m2
    Birth_plan_for_sow_update1(i,1)=i;
end
Birth_plan_for_sow_update1(1:7,2)=1;
Birth_plan_for_sow_update1(1:3,3)=4;

```

```

Birth_plan_for_sow_update1(4:7,3)=3;
Birth_plan_for_sow_update1(1:2,4)=1;
Birth_plan_for_sow_update1(1:2,5)=3;
Birth_plan_for_sow_update1(3:7,5)=2;

Birth_plan_for_sow_update2=zeros(700,7);
m1=size(Birth_plan_for_sow_update2,1);
for i=1:1:m1
    Birth_plan_for_sow_update2(i,1)=i;
end
Birth_plan_for_sow_update2(305:310,2)=2;
Birth_plan_for_sow_update2(311:364,2)=1;
Birth_plan_for_sow_update2(305:314,3)=2;
Birth_plan_for_sow_update2(315:364,3)=3;
Birth_plan_for_sow_update2(305:306,4)=1;
Birth_plan_for_sow_update2(317:364,5)=1;
%从后备母猪筛选
if day<10 && Birth_plan_for_sow_update1(day,kind1)>0
    aa=Birth_plan_for_sow_update1(day,kind1);
    A_sow_need_replace=Detail_sow_need_replace(A,kind);
    A_sow_need_replace=sortrows(A_sow_need_replace,-3);
    A_sow_pre=Detail_sow_pre(A,kind);
    A_sow_pre=sort(A_sow_pre,3);
    for j=1:1:aa
        bb=A_sow_pre(j,1);
        A(bb,5)=0;
        cc=A_sow_need_replace(j,1);
        A(cc,5)=5;
        A(cc,6)=2;
    end
end

%从育肥期母猪筛选
if day>100 && day<365
    if Birth_plan_for_sow_update2(day,kind1)>0

        aa= Birth_plan_for_sow_update2(day,kind1);
        A_sow_need_replace=Detail_sow_need_replace(A,kind);
        A_sow_need_replace=sortrows(A_sow_need_replace,-3);
        day_num=zeros(aa,1);
        for j=1:1:aa
            day_num(j,1)=A_sow_need_replace(j,3);
            bb=A_sow_need_replace(j,1);
            A(bb,3)=217+rand(1)*30;
        end

        A_finish_sow_to_replace=Detail_finish_sow(A,kind,aa);

        A_finish_sow_to_replace=sortrows(A_finish_sow_to_replace,-3);
        for j=1:1:aa

```

```

        bb=A_finish_sow_to_replace(j,1);
        if bb >0
            A(bb,3)=day_num(j);
        end
    end
end
end
end

```

对后备母猪进行更新函数

```

function A=Replace_sow_pre(A,kind,day)
    m=size(A,1);
    if day>10 && day<100
        for i=1:1:m
            if abs(A(i,2)-kind)<0.00001 && abs(A(i,4)-0)<0.00001 &&
abs(A(i,5)-2)<0.00001
                A(i,5)=5;
                A(i,6)=2;
            end
        end
    end
end

```

```

    kind1=kind+1;
    Birth_plan_for_sow_update2=zeros(631,7);
    m1=size(Birth_plan_for_sow_update2,1);
    for i=1:1:m1
        Birth_plan_for_sow_update2(i,1)=i;
    end
    Birth_plan_for_sow_update2(358:364,2)=1;
    Birth_plan_for_sow_update2(358:360,3)=4;
    Birth_plan_for_sow_update2(361:364,3)=3;
    Birth_plan_for_sow_update2(358:364,4)=1;
    Birth_plan_for_sow_update2(358:359,5)=3;
    Birth_plan_for_sow_update2(360:364,5)=2;

    if day>100 && Birth_plan_for_sow_update2(day,kind1)>0

```

```

        aa=Birth_plan_for_sow_update2(day,kind1);
        A_finish_sow_to_replace=Detail_finish_sow(A,kind,aa+2);
        A_finish_sow_to_replace=sortrows(A_finish_sow_to_replace,-3);
        for j=1:1:aa
            bb=A_finish_sow_to_replace(j,1);
            A(bb,3)=217+rand(1)*10;
            A(bb,5)=2;
            A(bb,6)=1;
        end
    end
end

```

```

end

```

```

end
end

```

对保育期小猪进行更新函数

```

function B=nurse_update(A,kind,date)
m=size(A,1);

for i=1:1:m
    if abs(A(i,3)-date)<0.01 && abs(A(i,2)-kind)<0.01 && abs(A(i,7)-1)<0.01 &&
abs(A(i,5)-8)>0.5
        A(i,5)=4;
    end
end
num_sow=0;
num_boar=0;
for i=1:1:m
    if abs(A(i,3)-date)<0.1 && abs(A(i,4)-0)<0.01 && abs(A(i,2)-kind)<0.01 &&
abs(A(i,7)-0)<0.01&& abs(A(i,5)-8)>0.5
        num_sow=num_sow+1;
        A(i,5)=4;
        A_sow_for_nurse_point(num_sow,:)=A(i,:);
    end
    if abs(A(i,3)-date)<0.1 && abs(A(i,4)-1)<0.01 && abs(A(i,2)-kind)<0.01 &&
abs(A(i,7)-0)<0.01 && abs(A(i,5)-8)>0.5
        num_boar=num_boar+1;
        A(i,5)=4;
        A_boar_for_nurse_point(num_boar,:)=A(i,:);
    end
end
num_sow_cut=round(0.03*num_sow);
num_boar_cut=round(0.03*num_boar);
num_cut=num_sow_cut+num_boar_cut;
aa=zeros(num_cut,1);
if num_sow_cut>0.1
    for i=1:1:num_sow_cut
        aa(i)=A_sow_for_nurse_point(i,1);
    end
end
if num_boar_cut>0.1
    for i=num_sow_cut+1:1:num_sow_cut+num_boar_cut
        aa(i)=A_boar_for_nurse_point(i,1);
    end
end
if num_cut>0.1
    A(aa,:)=[];          %%删除死亡的小猪的信息
end
m=size(A,1);
for i=1:1:m
    A(i,1)=i;

```

```

end
B=A;
end

```

对育肥期/待售猪进行更新函数

```

function A=finish_update(A,kind,date)
m=size(A,1);
for i=1:1:m
    if abs(A(i,3)-date)<0.01 && abs(A(i,4)-0)<0.01 && abs(A(i,2)-kind)<0.01 &&
abs(A(i,7)-1)<0.01 && abs(A(i,5)-8)>0.5
        A(i,5)=6;
    end
    if abs(A(i,3)-date)<0.01 && abs(A(i,4)-1)<0.01 && abs(A(i,2)-kind)<0.01 &&
abs(A(i,7)-1)<0.01 && abs(A(i,5)-8)>0.5
        A(i,5)=6;
    end
end

num_sow=0;
num_boar=0;
for i=1:1:m
    if abs(A(i,3)-date)<0.01 && abs(A(i,4)-0)<0.01 && abs(A(i,2)-kind)<0.01 &&
abs(A(i,7)-0)<0.01 && abs(A(i,5)-8)>0.5
        num_sow=num_sow+1;
        A(i,5)=5;
        A_sow_for_nurse_point(num_sow,:)=A(i,:);
    end
    if abs(A(i,3)-date)<0.01 && abs(A(i,4)-1)<0.01 && abs(A(i,2)-kind)<0.01 &&
abs(A(i,7)-0)<0.01&& abs(A(i,5)-8)>0.5
        num_boar=num_boar+1;
        A(i,5)=5;
        A_boar_for_nurse_point(num_boar,:)=A(i,:);
    end
end
num_sow_cut=round(0.02*num_sow);
num_boar_cut=round(0.02*num_boar);
num_cut=num_sow_cut+num_boar_cut;
aa=zeros(num_cut,1);
if num_sow_cut>0.1
    for i=1:1:num_sow_cut
        aa(i)=A_sow_for_nurse_point(i,1);
    end
end
if num_boar_cut>0.1
    for i=num_sow_cut+1:1:num_sow_cut+num_boar_cut
        aa(i)=A_boar_for_nurse_point(i,1);
    end
end
end

```

```

if num_cut>0.1
    A(aa,:)=[];          %%删除死亡的小猪的信息
end
m=size(A,1);
for i=1:1:m
    A(i,1)=i;
end
for i=1:1:m
    if abs(A(i,3)-date)<0.01 && abs(A(i,4)-0)<0.01 && abs(A(i,2)-kind)<0.01 &&
abs(A(i,6)-1)<0.01 && abs(A(i,5)-8)>0.5
        A(i,5)=6;
    end
    if abs(A(i,3)-date)<0.01 && abs(A(i,4)-1)<0.01 && abs(A(i,2)-kind)<0.01 &&
abs(A(i,6)-1)<0.01 && abs(A(i,5)-8)>0.5
        A(i,5)=6;
    end
end
end

end

```

对种猪进行选种函数

对母猪种猪进行选种

```

function A=Select_boar_last(A,kind,sex,select_date,percentage)
select_date=select_date-1;
[m,~]=size(A);
for i=1:1:m
    if abs(A(i,2)-kind)<0.1 && abs(A(i,3)-select_date)<0.1 && abs(A(i,4)-sex)<0.1 &&
abs(A(i,6)-1)<0.1 && abs(A(i,7)-1)<0.1 && abs(A(i,5)-8)>0.5
        A(i,5)=7;
    end
end

end

num=0;
for i=1:1:m
    if abs(A(i,2)-kind)<0.1 && abs(A(i,3)-select_date)<0.1 && abs(A(i,4)-sex)<0.1&&
abs(A(i,6)-1)<0.1 && abs(A(i,7)-0)<0.1 && abs(A(i,5)-8)>0.5
        num=num+1;
    end
end

end
select_num=round(percentage*num);
num2=0;
for i=1:1:m
    if abs(A(i,2)-kind)<0.1 && abs(A(i,3)-select_date)<0.1 && abs(A(i,4)-sex)<0.1&&
abs(A(i,6)-1)<0.1 && abs(A(i,7)-0)<0.1 && abs(A(i,5)-8)>0.5
        if num2<select_num

```

```

        num2=num2+1;
        A(i,5)=7;
    end
end
if num2>select_num-0.5
    break
end
end
num2=0;
for i=1:1:m
    if abs(A(i,2)-kind)<0.1 && abs(A(i,3)-select_date)<0.1 && abs(A(i,4)-sex)<0.1&&
abs(A(i,6)-1)<0.1 && abs(A(i,7)-0)<0.1&& abs(A(i,5)-8)>0.5
        if num2>select_num-0.5
            num2=num2+1;
            A(i,5)=5;
            A(i,6)=2;
        end
    end
end
end
end
对公猪种猪进行选种

```

```

function A=Select_sow_last(A,kind,sex,select_date,percentage)
select_date=select_date-1;
[m,~]=size(A);

for i=1:1:m
    if abs(A(i,2)-kind)<0.1 && abs(A(i,3)-select_date)<0.1 && abs(A(i,4)-sex)<0.1 &&
abs(A(i,6)-1)<0.1 && abs(A(i,7)-1)<0.1&& abs(A(i,5)-8)>0.5
        A(i,5)=7;
    end

end
num=0;
for i=1:1:m
    if abs(A(i,2)-kind)<0.1 && abs(A(i,3)-select_date)<0.1 && abs(A(i,4)-sex)<0.1 &&
abs(A(i,6)-1)<0.1 && abs(A(i,7)-0)<0.1&& abs(A(i,5)-8)>0.5
        num=num+1;
    end

end
select_num=round(percentage*num);
num2=0;
for i=1:1:m
    if abs(A(i,2)-kind)<0.1 && abs(A(i,3)-select_date)<0.1 && abs(A(i,4)-sex)<0.1&&
abs(A(i,6)-1)<0.1 && abs(A(i,7)-0)<0.1 && abs(A(i,5)-8)>0.5
        if num2<select_num
            num2=num2+1;
            A(i,5)=7;

```

```

        end
    end
    if num2>select_num-0.5
        break
    end
end
num2=0;
for i=1:1:m
    if abs(A(i,2)-kind)<0.1 && abs(A(i,3)-select_date)<0.1 && abs(A(i,4)-sex)<0.1&&
abs(A(i,6)-1)<0.1 && abs(A(i,7)-0)<0.1&& abs(A(i,5)-8)>0.5
        if num2>select_num-0.5
            num2=num2+1;
            A(i,5)=5;
            A(i,6)=2;
        end
    end
end
end
end
end

```

对副产品哺乳仔猪个数统计函数

```

function num_for_child=number_for_child_last(Origin_num_data)
[m,~]=size(Origin_num_data);
num_A_sow_for_child=0;
num_A_boar_for_child=0;
num_B_sow_for_child=0;
num_B_boar_for_child=0;
num_C_sow_for_child=0;
num_C_boar_for_child=0;
num_D_sow_for_child=0;
num_D_boar_for_child=0;
num_E_sow_for_child=0;
num_E_boar_for_child=0;
num_F_sow_for_child=0;
num_F_boar_for_child=0;
for i=1:1:m
    if abs(Origin_num_data(i,2)-1)<0.00001 && abs(Origin_num_data(i,4)-0)<0.00001
&& abs(Origin_num_data(i,5)-3)<0.00001 && abs(Origin_num_data(i,6)-2)<0.00001
        num_A_sow_for_child=num_A_sow_for_child+1;
        % A_sow_for_child( num_A_sow_for_child,:)=Origin_num_data(i,:);
    end
    if abs(Origin_num_data(i,2)-1)<0.00001 && abs(Origin_num_data(i,4)-1)<0.00001
&& abs(Origin_num_data(i,5)-3)<0.00001 && abs(Origin_num_data(i,6)-2)<0.00001
        num_A_boar_for_child=num_A_boar_for_child+1;
        % A_boar_for_child( num_A_boar_for_child,:)=Origin_num_data(i,:);
    end
    if
        abs(Origin_num_data(i,2)-2)<0.00001
&& abs(Origin_num_data(i,4)-0)<0.00001 && abs(Origin_num_data(i,5)-3)<0.00001 &&
abs(Origin_num_data(i,6)-2)<0.00001
        num_B_sow_for_child=num_B_sow_for_child+1;
        % B_sow_for_child( num_B_sow_for_child,:)=Origin_num_data(i,:);
    end
end
end

```

```

end
if abs(Origin_num_data(i,2)-2)<0.00001 && abs(Origin_num_data(i,4)-1)<0.00001
&& abs(Origin_num_data(i,5)-3)<0.00001 && abs(Origin_num_data(i,6)-2)<0.00001
    num_B_boar_for_child=num_B_boar_for_child+1;
    % B_boar_for_child( num_B_boar_for_child,:)=Origin_num_data(i,:);
end
if
    abs(Origin_num_data(i,2)-3)<0.00001 &&
abs(Origin_num_data(i,4)-0)<0.00001 && abs(Origin_num_data(i,5)-3)<0.00001 &&
abs(Origin_num_data(i,6)-2)<0.00001
    num_C_sow_for_child=num_C_sow_for_child+1;
    % C_sow_for_child( num_C_sow_for_child,:)=Origin_num_data(i,:);
end
if abs(Origin_num_data(i,2)-3)<0.00001 && abs(Origin_num_data(i,4)-1)<0.00001
&& abs(Origin_num_data(i,5)-3)<0.00001 && abs(Origin_num_data(i,6)-2)<0.00001
    num_C_boar_for_child=num_C_boar_for_child+1;
    %C_boar_for_child( num_C_boar_for_child,:)=Origin_num_data(i,:);
end
if
    abs(Origin_num_data(i,2)-4)<0.00001 &&
abs(Origin_num_data(i,4)-0)<0.00001 && abs(Origin_num_data(i,5)-3)<0.00001 &&
abs(Origin_num_data(i,6)-2)<0.00001
    num_D_sow_for_child=num_D_sow_for_child+1;
    %D_sow_for_child( num_D_sow_for_child,:)=Origin_num_data(i,:);
end
if abs(Origin_num_data(i,2)-4)<0.00001 && abs(Origin_num_data(i,4)-1)<0.00001
&& abs(Origin_num_data(i,5)-3)<0.00001 && abs(Origin_num_data(i,6)-2)<0.00001
    num_D_boar_for_child=num_D_boar_for_child+1;
    % D_boar_for_child( num_D_boar_for_child,:)=Origin_num_data(i,:);
end
if abs(Origin_num_data(i,2)-5)<0.00001 && abs(Origin_num_data(i,4)-0)<0.00001
&& abs(Origin_num_data(i,5)-3)<0.00001 && abs(Origin_num_data(i,6)-2)<0.00001
    num_E_sow_for_child=num_E_sow_for_child+1;
    % E_sow_for_child( num_E_sow_for_child,:)=Origin_num_data(i,:);
end
if abs(Origin_num_data(i,2)-5)<0.00001 && abs(Origin_num_data(i,4)-1)<0.00001
&& abs(Origin_num_data(i,5)-3)<0.00001 && abs(Origin_num_data(i,6)-2)<0.00001
    num_E_boar_for_child=num_E_boar_for_child+1;
    % E_boar_for_child( num_E_boar_for_child,:)=Origin_num_data(i,:);
end
if abs(Origin_num_data(i,2)-6)<0.00001 && abs(Origin_num_data(i,4)-0)<0.00001
&& abs(Origin_num_data(i,5)-3)<0.00001 && abs(Origin_num_data(i,6)-2)<0.00001
    num_F_sow_for_child=num_F_sow_for_child+1;
    % F_sow_for_child( num_F_sow_for_child,:)=Origin_num_data(i,:);
end
if abs(Origin_num_data(i,2)-6)<0.00001 && abs(Origin_num_data(i,4)-1)<0.00001
&& abs(Origin_num_data(i,5)-3)<0.00001 && abs(Origin_num_data(i,6)-2)<0.00001
    num_F_boar_for_child=num_F_boar_for_child+1;
    % F_boar_for_child( num_F_boar_for_child,:)=Origin_num_data(i,:);
end
end
num_for_child(1,1)=num_A_sow_for_child;

```

```

num_for_child(1,2)=num_A_boar_for_child;
num_for_child(2,1)=num_B_sow_for_child;
num_for_child(2,2)=num_B_boar_for_child;
num_for_child(3,1)=num_C_sow_for_child;
num_for_child(3,2)=num_C_boar_for_child;
num_for_child(4,1)=num_D_sow_for_child;
num_for_child(4,2)=num_D_boar_for_child;
num_for_child(5,1)=num_E_sow_for_child;
num_for_child(5,2)=num_E_boar_for_child;
num_for_child(6,1)=num_F_sow_for_child;
num_for_child(6,2)=num_F_boar_for_child;
num_for_child(7,1)=num_A_sow_for_child+num_B_sow_for_child+num_C_sow_for_c
hild+num_D_sow_for_child+num_E_sow_for_child+num_F_sow_for_child+num_A_boar_f
or_child+num_B_boar_for_child+num_C_boar_for_child+num_D_boar_for_child+num_E_b
oar_for_child+num_F_boar_for_child;
num_for_child(7,2)=num_for_child(7,1);
end
%num_for_child=num_A_sow_for_child+num_B_sow_for_child+num_C_sow_for_chil
d+num_D_sow_for_child+num_E_sow_for_child+num_F_sow_for_child+num_A_boar_for_
child+num_B_boar_for_child+num_C_boar_for_child+num_D_boar_for_child+num_E_boar
_for_child+num_F_boar_for_child;

```

对副产品保育猪个数统计函数

```

function num_for_nurse=number_for_nurse_last(Origin_num_data)
[m,~]=size(Origin_num_data);
num_A_sow_for_nurse=0;
num_A_boar_for_nurse=0;
num_B_sow_for_nurse=0;
num_B_boar_for_nurse=0;
num_C_sow_for_nurse=0;
num_C_boar_for_nurse=0;
num_D_sow_for_nurse=0;
num_D_boar_for_nurse=0;
num_E_sow_for_nurse=0;
num_E_boar_for_nurse=0;
num_F_sow_for_nurse=0;
num_F_boar_for_nurse=0;
for i=1:1:m
    if abs(Origin_num_data(i,2)-1)<0.00001 && abs(Origin_num_data(i,4)-0)<0.00001
    && abs(Origin_num_data(i,5)-4)<0.00001 && abs(Origin_num_data(i,6)-2)<0.00001
        num_A_sow_for_nurse=num_A_sow_for_nurse+1;
        % A_sow_for_nurse( num_A_sow_for_nurse,:)=Origin_num_data(i,:);
    end
    if abs(Origin_num_data(i,2)-1)<0.00001 && abs(Origin_num_data(i,4)-1)<0.00001
    && abs(Origin_num_data(i,5)-4)<0.00001 && abs(Origin_num_data(i,6)-2)<0.00001
        num_A_boar_for_nurse=num_A_boar_for_nurse+1;
        % A_boar_for_nurse( num_A_boar_for_nurse,:)=Origin_num_data(i,:);
    end
end

```

```

        if abs(Origin_num_data(i,2)-2)<0.00001 %&&
abs(Origin_num_data(i,4)-0)<0.00001 %&& abs(Origin_num_data(i,5)-4)<0.00001 %&&
abs(Origin_num_data(i,6)-2)<0.00001
            num_B_sow_for_nurse=num_B_sow_for_nurse+1;
            % B_sow_for_nurse( num_B_sow_for_nurse,:)=Origin_num_data(i,:);
        end
        if abs(Origin_num_data(i,2)-2)<0.00001 %&& abs(Origin_num_data(i,4)-1)<0.00001
%&& abs(Origin_num_data(i,5)-4)<0.00001 %&& abs(Origin_num_data(i,6)-2)<0.00001
            num_B_boar_for_nurse=num_B_boar_for_nurse+1;
            % B_boar_for_nurse( num_B_boar_for_nurse,:)=Origin_num_data(i,:);
        end
        if abs(Origin_num_data(i,2)-3)<0.00001 %&&
abs(Origin_num_data(i,4)-0)<0.00001 %&& abs(Origin_num_data(i,5)-4)<0.00001 %&&
abs(Origin_num_data(i,6)-2)<0.00001
            num_C_sow_for_nurse=num_C_sow_for_nurse+1;
            % C_sow_for_nurse( num_C_sow_for_nurse,:)=Origin_num_data(i,:);
        end
        if abs(Origin_num_data(i,2)-3)<0.00001 %&& abs(Origin_num_data(i,4)-1)<0.00001
%&& abs(Origin_num_data(i,5)-4)<0.00001 %&& abs(Origin_num_data(i,6)-2)<0.00001
            num_C_boar_for_nurse=num_C_boar_for_nurse+1;
            %C_boar_for_nurse( num_C_boar_for_nurse,:)=Origin_num_data(i,:);
        end
        if abs(Origin_num_data(i,2)-4)<0.00001 %&&
abs(Origin_num_data(i,4)-0)<0.00001 %&& abs(Origin_num_data(i,5)-4)<0.00001 %&&
abs(Origin_num_data(i,6)-2)<0.00001
            num_D_sow_for_nurse=num_D_sow_for_nurse+1;
            %D_sow_for_nurse( num_D_sow_for_nurse,:)=Origin_num_data(i,:);
        end
        if abs(Origin_num_data(i,2)-4)<0.00001 %&& abs(Origin_num_data(i,4)-1)<0.00001
%&& abs(Origin_num_data(i,5)-4)<0.00001 %&& abs(Origin_num_data(i,6)-2)<0.00001
            num_D_boar_for_nurse=num_D_boar_for_nurse+1;
            % D_boar_for_nurse( num_D_boar_for_nurse,:)=Origin_num_data(i,:);
        end
        if abs(Origin_num_data(i,2)-5)<0.00001 %&& abs(Origin_num_data(i,4)-0)<0.00001
%&& abs(Origin_num_data(i,5)-4)<0.00001 %&& abs(Origin_num_data(i,6)-2)<0.00001
            num_E_sow_for_nurse=num_E_sow_for_nurse+1;
            % E_sow_for_nurse( num_E_sow_for_nurse,:)=Origin_num_data(i,:);
        end
        if abs(Origin_num_data(i,2)-5)<0.00001 %&& abs(Origin_num_data(i,4)-1)<0.00001
%&& abs(Origin_num_data(i,5)-4)<0.00001 %&& abs(Origin_num_data(i,6)-2)<0.00001
            num_E_boar_for_nurse=num_E_boar_for_nurse+1;
            % E_boar_for_nurse( num_E_boar_for_nurse,:)=Origin_num_data(i,:);
        end
        if abs(Origin_num_data(i,2)-6)<0.00001 %&& abs(Origin_num_data(i,4)-0)<0.00001
%&& abs(Origin_num_data(i,5)-4)<0.00001 %&& abs(Origin_num_data(i,6)-2)<0.00001
            num_F_sow_for_nurse=num_F_sow_for_nurse+1;
            % F_sow_for_nurse( num_F_sow_for_nurse,:)=Origin_num_data(i,:);
        end
        if abs(Origin_num_data(i,2)-6)<0.00001 %&& abs(Origin_num_data(i,4)-1)<0.00001
%&& abs(Origin_num_data(i,5)-4)<0.00001 %&& abs(Origin_num_data(i,6)-2)<0.00001

```

```

        num_F_boar_for_nurse=num_F_boar_for_nurse+1;
%        F_boar_for_nurse( num_F_boar_for_nurse,:)=Origin_num_data(i,:);
    end
end
num_for_nurse(1,1)=num_A_sow_for_nurse;
num_for_nurse(1,2)=num_A_boar_for_nurse;
num_for_nurse(2,1)=num_B_sow_for_nurse;
num_for_nurse(2,2)=num_B_boar_for_nurse;
num_for_nurse(3,1)=num_C_sow_for_nurse;
num_for_nurse(3,2)=num_C_boar_for_nurse;
num_for_nurse(4,1)=num_D_sow_for_nurse;
num_for_nurse(4,2)=num_D_boar_for_nurse;
num_for_nurse(5,1)=num_E_sow_for_nurse;
num_for_nurse(5,2)=num_E_boar_for_nurse;
num_for_nurse(6,1)=num_F_sow_for_nurse;
num_for_nurse(6,2)=num_F_boar_for_nurse;
num_for_nurse(7,1)=num_A_sow_for_nurse+num_B_sow_for_nurse+num_C_sow_for_nurse+num_D_sow_for_nurse+num_E_sow_for_nurse+num_F_sow_for_nurse+num_A_boar_for_nurse+num_B_boar_for_nurse+num_C_boar_for_nurse+num_D_boar_for_nurse+num_E_boar_for_nurse+num_F_boar_for_nurse;
num_for_nurse(7,2)=num_for_nurse(7,1);
end
%num_for_nurse=num_A_sow_for_nurse+num_B_sow_for_nurse+num_C_sow_for_nurse+num_D_sow_for_nurse+num_E_sow_for_nurse+num_F_sow_for_nurse+num_A_boar_for_nurse+num_B_boar_for_nurse+num_C_boar_for_nurse+num_D_boar_for_nurse+num_E_boar_for_nurse+num_F_boar_for_nurse;

```

对副产品育肥猪个数统计函数

```

function num_for_finish=number_for_finish_last(Origin_num_data)
[m,~]=size(Origin_num_data);
num_A_sow_for_finish=0;
num_A_boar_for_finish=0;
num_B_sow_for_finish=0;
num_B_boar_for_finish=0;
num_C_sow_for_finish=0;
num_C_boar_for_finish=0;
num_D_sow_for_finish=0;
num_D_boar_for_finish=0;
num_E_sow_for_finish=0;
num_E_boar_for_finish=0;
num_F_sow_for_finish=0;
num_F_boar_for_finish=0;
for i=1:1:m
    if abs(Origin_num_data(i,2)-1)<0.00001 && abs(Origin_num_data(i,4)-0)<0.00001 && abs(Origin_num_data(i,5)-5)<0.00001 && abs(Origin_num_data(i,6)-2)<0.00001
        num_A_sow_for_finish=num_A_sow_for_finish+1;
        % A_sow_for_finish( num_A_sow_for_finish,:)=Origin_num_data(i,:);
    end
    if abs(Origin_num_data(i,2)-1)<0.00001 && abs(Origin_num_data(i,4)-1)<0.00001 && abs(Origin_num_data(i,5)-5)<0.00001 && abs(Origin_num_data(i,6)-2)<0.00001

```

```

        num_A_boar_for_finish=num_A_boar_for_finish+1;
        % A_boar_for_finish( num_A_boar_for_finish,:)=Origin_num_data(i,:);
    end
    if          abs(Origin_num_data(i,2)-2)<0.00001          &&
abs(Origin_num_data(i,4)-0)<0.00001  &&  abs(Origin_num_data(i,5)-5)<0.00001  &&
abs(Origin_num_data(i,6)-2)<0.00001
        num_B_sow_for_finish=num_B_sow_for_finish+1;
        % B_sow_for_finish( num_B_sow_for_finish,:)=Origin_num_data(i,:);
    end
    if abs(Origin_num_data(i,2)-2)<0.00001 && abs(Origin_num_data(i,4)-1)<0.00001
&& abs(Origin_num_data(i,5)-5)<0.00001 && abs(Origin_num_data(i,6)-2)<0.00001
        num_B_boar_for_finish=num_B_boar_for_finish+1;
        % B_boar_for_finish( num_B_boar_for_finish,:)=Origin_num_data(i,:);
    end
    if          abs(Origin_num_data(i,2)-3)<0.00001          &&
abs(Origin_num_data(i,4)-0)<0.00001  &&  abs(Origin_num_data(i,5)-5)<0.00001  &&
abs(Origin_num_data(i,6)-2)<0.00001
        num_C_sow_for_finish=num_C_sow_for_finish+1;
        % C_sow_for_finish( num_C_sow_for_finish,:)=Origin_num_data(i,:);
    end
    if abs(Origin_num_data(i,2)-3)<0.00001 && abs(Origin_num_data(i,4)-1)<0.00001
&& abs(Origin_num_data(i,5)-5)<0.00001 && abs(Origin_num_data(i,6)-2)<0.00001
        num_C_boar_for_finish=num_C_boar_for_finish+1;
        %C_boar_for_finish( num_C_boar_for_finish,:)=Origin_num_data(i,:);
    end
    if          abs(Origin_num_data(i,2)-4)<0.00001          &&
abs(Origin_num_data(i,4)-0)<0.00001  &&  abs(Origin_num_data(i,5)-5)<0.00001  &&
abs(Origin_num_data(i,6)-2)<0.00001
        num_D_sow_for_finish=num_D_sow_for_finish+1;
        %D_sow_for_finish( num_D_sow_for_finish,:)=Origin_num_data(i,:);
    end
    if abs(Origin_num_data(i,2)-4)<0.00001 && abs(Origin_num_data(i,4)-1)<0.00001
&& abs(Origin_num_data(i,5)-5)<0.00001 && abs(Origin_num_data(i,6)-2)<0.00001
        num_D_boar_for_finish=num_D_boar_for_finish+1;
        % D_boar_for_finish( num_D_boar_for_finish,:)=Origin_num_data(i,:);
    end
    if abs(Origin_num_data(i,2)-5)<0.00001 && abs(Origin_num_data(i,4)-0)<0.00001
&& abs(Origin_num_data(i,5)-5)<0.00001 && abs(Origin_num_data(i,6)-2)<0.00001
        num_E_sow_for_finish=num_E_sow_for_finish+1;
        %      E_sow_for_finish( num_E_sow_for_finish,:)=Origin_num_data(i,:);
    end
    if abs(Origin_num_data(i,2)-5)<0.00001 && abs(Origin_num_data(i,4)-1)<0.00001
&& abs(Origin_num_data(i,5)-5)<0.00001 && abs(Origin_num_data(i,6)-2)<0.00001
        num_E_boar_for_finish=num_E_boar_for_finish+1;
        %      E_boar_for_finish( num_E_boar_for_finish,:)=Origin_num_data(i,:);
    end
    if abs(Origin_num_data(i,2)-6)<0.00001 && abs(Origin_num_data(i,4)-0)<0.00001
&& abs(Origin_num_data(i,5)-5)<0.00001 && abs(Origin_num_data(i,6)-2)<0.00001
        num_F_sow_for_finish=num_F_sow_for_finish+1;
        %      F_sow_for_finish( num_F_sow_for_finish,:)=Origin_num_data(i,:);

```

```

        end
        if abs(Origin_num_data(i,2)-6)<0.00001 && abs(Origin_num_data(i,4)-1)<0.00001
        && abs(Origin_num_data(i,5)-5)<0.00001 && abs(Origin_num_data(i,6)-2)<0.00001
            num_F_boar_for_finish=num_F_boar_for_finish+1;
            % F_boar_for_finish( num_F_boar_for_finish,:)=Origin_num_data(i,:);
        end
    end
    num_for_finish(1,1)=num_A_sow_for_finish;
    num_for_finish(1,2)=num_A_boar_for_finish;
    num_for_finish(2,1)=num_B_sow_for_finish;
    num_for_finish(2,2)=num_B_boar_for_finish;
    num_for_finish(3,1)=num_C_sow_for_finish;
    num_for_finish(3,2)=num_C_boar_for_finish;
    num_for_finish(4,1)=num_D_sow_for_finish;
    num_for_finish(4,2)=num_D_boar_for_finish;
    num_for_finish(5,1)=num_E_sow_for_finish;
    num_for_finish(5,2)=num_E_boar_for_finish;
    num_for_finish(6,1)=num_F_sow_for_finish;
    num_for_finish(6,2)=num_F_boar_for_finish;
    num_for_finish(7,1)=num_A_sow_for_finish+num_B_sow_for_finish+num_C_sow_for_
_finish+num_D_sow_for_finish+num_E_sow_for_finish+num_F_sow_for_finish+num_A_b
oar_for_finish+num_B_boar_for_finish+num_C_boar_for_finish+num_D_boar_for_finish+n
um_E_boar_for_finish+num_F_boar_for_finish;
    num_for_finish(7,2)=num_for_finish(7,1);
    end
    %num_for_finish=num_A_sow_for_finish+num_B_sow_for_finish+num_C_sow_for_fi
nish+num_D_sow_for_finish+num_E_sow_for_finish+num_F_sow_for_finish+num_A_boar
_for_finish+num_B_boar_for_finish+num_C_boar_for_finish+num_D_boar_for_finish+num
_E_boar_for_finish+num_F_boar_for_finish;

```

对可卖种猪个数统计函数

```

function num_for_sale_real=number_for_sale_real(Origin_num_data) %可卖种猪的
数量
[m,~]=size(Origin_num_data);
num_A_sow_for_sale_real=0;
num_A_boar_for_sale_real=0;
num_B_sow_for_sale_real=0;
num_B_boar_for_sale_real=0;
num_C_sow_for_sale_real=0;
num_C_boar_for_sale_real=0;
num_D_sow_for_sale_real=0;
num_D_boar_for_sale_real=0;
num_E_sow_for_sale_real=0;
num_E_boar_for_sale_real=0;
num_F_sow_for_sale_real=0;
num_F_boar_for_sale_real=0;
for i=1:1:m
    if abs(Origin_num_data(i,2)-1)<0.00001 && abs(Origin_num_data(i,4)-0)<0.00001
    && abs(Origin_num_data(i,5)-7)<0.00001 && abs(Origin_num_data(i,6)-1)<0.00001
        num_A_sow_for_sale_real=num_A_sow_for_sale_real+1;
    end
end

```

```

% A_sow_for_sale_real( num_A_sow_for_sale_real,:)=Origin_num_data(i,:);
end
if abs(Origin_num_data(i,2)-1)<0.00001 && abs(Origin_num_data(i,4)-1)<0.00001
&& abs(Origin_num_data(i,5)-7)<0.00001 && abs(Origin_num_data(i,6)-1)<0.00001
    num_A_boar_for_sale_real=num_A_boar_for_sale_real+1;
    % A_boar_for_sale_real( num_A_boar_for_sale_real,:)=Origin_num_data(i,:);
end
if
    abs(Origin_num_data(i,2)-2)<0.00001 &&
abs(Origin_num_data(i,4)-0)<0.00001 && abs(Origin_num_data(i,5)-7)<0.00001 &&
abs(Origin_num_data(i,6)-1)<0.00001
    num_B_sow_for_sale_real=num_B_sow_for_sale_real+1;
    % B_sow_for_sale_real( num_B_sow_for_sale_real,:)=Origin_num_data(i,:);
end
if abs(Origin_num_data(i,2)-2)<0.00001 && abs(Origin_num_data(i,4)-1)<0.00001
&& abs(Origin_num_data(i,5)-7)<0.00001 && abs(Origin_num_data(i,6)-1)<0.00001
    num_B_boar_for_sale_real=num_B_boar_for_sale_real+1;
    % B_boar_for_sale_real( num_B_boar_for_sale_real,:)=Origin_num_data(i,:);
end
if
    abs(Origin_num_data(i,2)-3)<0.00001 &&
abs(Origin_num_data(i,4)-0)<0.00001 && abs(Origin_num_data(i,5)-7)<0.00001 &&
abs(Origin_num_data(i,6)-1)<0.00001
    num_C_sow_for_sale_real=num_C_sow_for_sale_real+1;
    % C_sow_for_sale_real( num_C_sow_for_sale_real,:)=Origin_num_data(i,:);
end
if abs(Origin_num_data(i,2)-3)<0.00001 && abs(Origin_num_data(i,4)-1)<0.00001
&& abs(Origin_num_data(i,5)-7)<0.00001 && abs(Origin_num_data(i,6)-1)<0.00001
    num_C_boar_for_sale_real=num_C_boar_for_sale_real+1;
    %C_boar_for_sale_real( num_C_boar_for_sale_real,:)=Origin_num_data(i,:);
end
if
    abs(Origin_num_data(i,2)-4)<0.00001 &&
abs(Origin_num_data(i,4)-0)<0.00001 && abs(Origin_num_data(i,5)-7)<0.00001 &&
abs(Origin_num_data(i,6)-1)<0.00001
    num_D_sow_for_sale_real=num_D_sow_for_sale_real+1;
    %D_sow_for_sale_real( num_D_sow_for_sale_real,:)=Origin_num_data(i,:);
end
if abs(Origin_num_data(i,2)-4)<0.00001 && abs(Origin_num_data(i,4)-1)<0.00001
&& abs(Origin_num_data(i,5)-7)<0.00001 && abs(Origin_num_data(i,6)-1)<0.00001
    num_D_boar_for_sale_real=num_D_boar_for_sale_real+1;
    % D_boar_for_sale_real( num_D_boar_for_sale_real,:)=Origin_num_data(i,:);
end
if abs(Origin_num_data(i,2)-5)<0.00001 && abs(Origin_num_data(i,4)-0)<0.00001
&& abs(Origin_num_data(i,5)-7)<0.00001 && abs(Origin_num_data(i,6)-1)<0.00001
    num_E_sow_for_sale_real=num_E_sow_for_sale_real+1;
    % E_sow_for_sale_real( num_E_sow_for_sale_real,:)=Origin_num_data(i,:);
end
if abs(Origin_num_data(i,2)-5)<0.00001 && abs(Origin_num_data(i,4)-1)<0.00001
&& abs(Origin_num_data(i,5)-7)<0.00001 && abs(Origin_num_data(i,6)-1)<0.00001
    num_E_boar_for_sale_real=num_E_boar_for_sale_real+1;
    % E_boar_for_sale_real( num_E_boar_for_sale_real,:)=Origin_num_data(i,:);
end

```

```

        if abs(Origin_num_data(i,2)-6)<0.00001 && abs(Origin_num_data(i,4)-0)<0.00001
        && abs(Origin_num_data(i,5)-7)<0.00001 && abs(Origin_num_data(i,6)-1)<0.00001
            num_F_sow_for_sale_real=num_F_sow_for_sale_real+1;
        %           F_sow_for_sale_real( num_F_sow_for_sale_real,:)=Origin_num_data(i,:);
        end
        if abs(Origin_num_data(i,2)-6)<0.00001 && abs(Origin_num_data(i,4)-1)<0.00001
        && abs(Origin_num_data(i,5)-7)<0.00001 && abs(Origin_num_data(i,6)-1)<0.00001
            num_F_boar_for_sale_real=num_F_boar_for_sale_real+1;
        %           F_boar_for_sale_real( num_F_boar_for_sale_real,:)=Origin_num_data(i,:);
        end
    end
    num_for_sale_real(1,1)=num_A_sow_for_sale_real;
    num_for_sale_real(1,2)=num_A_boar_for_sale_real;
    num_for_sale_real(2,1)=num_B_sow_for_sale_real;
    num_for_sale_real(2,2)=num_B_boar_for_sale_real;
    num_for_sale_real(3,1)=num_C_sow_for_sale_real;
    num_for_sale_real(3,2)=num_C_boar_for_sale_real;
    num_for_sale_real(4,1)=num_D_sow_for_sale_real;
    num_for_sale_real(4,2)=num_D_boar_for_sale_real;
    num_for_sale_real(5,1)=num_E_sow_for_sale_real;
    num_for_sale_real(5,2)=num_E_boar_for_sale_real;
    num_for_sale_real(6,1)=num_F_sow_for_sale_real;
    num_for_sale_real(6,2)=num_F_boar_for_sale_real;
    num_for_sale_real(7,1)=num_A_sow_for_sale_real+num_B_sow_for_sale_real+num_C
_sow_for_sale_real+num_D_sow_for_sale_real+num_E_sow_for_sale_real+num_F_sow_for
_sale_real+num_A_boar_for_sale_real+num_B_boar_for_sale_real+num_C_boar_for_sale_r
eal+num_D_boar_for_sale_real+num_E_boar_for_sale_real+num_F_boar_for_sale_real;
    num_for_sale_real(7,2)=num_for_sale_real(7,1);
    end
    %num_for_sale_real=num_A_sow_for_sale_real+num_B_sow_for_sale_real+num_C_s
ow_for_sale_real+num_D_sow_for_sale_real+num_E_sow_for_sale_real+num_F_sow_for_s
ale_real+num_A_boar_for_sale_real+num_B_boar_for_sale_real+num_C_boar_for_sale_real
+num_D_boar_for_sale_real+num_E_boar_for_sale_real+num_F_boar_for_sale_real;

```

```

    %num_for_sale_real_real=num_A_sow_for_sale_real_real+num_B_sow_for_sale_real_
real+num_C_sow_for_sale_real_real+num_D_sow_for_sale_real_real+num_E_sow_for_sale
_real_real+num_F_sow_for_sale_real_real+num_A_boar_for_sale_real_real+num_B_boar_f
or_sale_real_real+num_C_boar_for_sale_real_real+num_D_boar_for_sale_real_real+num_E
_boar_for_sale_real_real+num_F_boar_for_sale_real_real;

```

对种猪售卖函数

```

function A=sale_zhong(A,kind,sex,num)
A_for_sale_real=Detail_for_sale_real(A,kind,sex);

A_for_sale_real=sortrows(A_for_sale_real,-3);

num1=size(A_for_sale_real,1);
aa=zeros(num,1);
for i=1:1:num

```

```

        aa(i)=A_for_sale_real(i,1);
    end
    A(aa,:)=[];

    m=size(A,1);
    for i=1:1:m
        A(i,1)=i;
    end
end
end

```

副产品销售淡季对副产品进行销售函数

```

function [Anew,saleplan]=poorseason(A,B)
% clc;
% A=load('before.txt');
% B=load('after.txt');
%% 上限信息
limits=1704;
limitn=2675;
limitf=5657;
1

%% 得到三种猪的详细信息及个数
small=A(find(A(:,5)==3),:);%初始猪的状态
nurse=A(find(A(:,5)==4),:);
fat=[A(find(A(:,5)==6),:);A(find(A(:,5)==5),:)];
smalla=B(find(B(:,5)==3),:);%一周后猪的状态
nursea=B(find(B(:,5)==4),:);
fata=[B(find(B(:,5)==6),:);B(find(B(:,5)==5),:)];
smallsale=small(find(small(:,6)==2),:);%可卖的副产品猪
nursesale=nurse(find(nurse(:,6)==2),:);
fatsale=fat(find(fat(:,6)==2),:);

%% 要卖的猪的数量
nsales=max(size(smalla,1)-limits,0);
nsalen=max(size(nursea,1)-limitn,0);
numberfat=fatsale(find(fatsale(:,3)>=201),:);
num=size(numberfat,1);%201 天以上的肥猪
nsalef1=max(size(fata,1)-limitf,0);
nsalef2=max(0,num);
nsalef=max(nsalef1,nsalef2);

%% 挑选要卖的猪仔
ss1=smallsale(find(smallsale(:,3)>=20),:);
ss1=sortrows(ss1,3);
ss=[];
if nsales>size(ss1,1)
    nsales=size(ss1,1);
end
% for i=1:nsales

```

```

%      ss=[ss;ss1(i,:)];
% end
if nsales~=0
    ss=selectpigs(smallsale,nsales);
end
% nsales=0;

%% 挑选要卖的保育猪
sn1=nursesale(find(nursesale(:,3)>=29),:);
sn1=sortrows(sn1,3);
sn=[];
% for i=1:nsalen
%      sn=[sn;sn1(i,:)];
% end
if nsalen~=0
    sn=selectpig(nursesale,nsalen);
end
%% 挑选要卖的育肥猪
sn111=numberfat;%大肥猪 201 天以上的猪
sf1=fatsale(find(fatsale(:,3)>=76),:);
sf1=sortrows(sf1,3);
sf=[];
% num=1;
for i=1:(nsalef-num)
    sf=[sf;sf1(i,:)];
end
sf=[sf;sn111];

%% 所有要卖的猪状态更新
totalsale=[ss;sn;sf];%仔猪，保育，育肥
Anew=A;
for i=1:size(totalsale,1)
    Anew(find(Anew(:,1)==totalsale(i,1)),:)=[];
end
for i=1:size(Anew,1)
    Anew(i,1)=i;
end

%% 本周销售计划输出
weight1=0;
count1=0;
weight2=0;
count2=0;
weight3=0;
count3=0;
weight4=0;
count4=0;
for i=1:size(totalsale,1)
    dweight=dw(totalsale(i,3));

```

```

if dweight<=20
    weight1=weight1+dweight;
    count1=count1+1;
elseif dweight>=25&dweight<=50
    weight2=weight2+dweight;
    count2=count2+1;
elseif dweight>=50&dweight<=100
    weight3=weight3+dweight;
    count3=count3+1;
elseif dweight>=100
    weight4=weight4+dweight;
    count4=count4+1;
end
end
count=count1+count2+count3+count4;%数据的简单核算
saleplan=[weight1,weight2,weight3,weight4];
% end

```

副产品销售旺季对副产品进行销售函数

```

function [Anew,saleplan]=richseason(A)
% clc;
% A=load('richtest.txt');

%% 下限信息
limit=9100;
naver=500;
limits=2050;
2

%% 得到三种猪的详细信息及个数
small=A(find(A(:,5)==3),:);%初始猪的状态
nurse=A(find(A(:,5)==4),:);
fat=[A(find(A(:,5)==6),:);A(find(A(:,5)==5),:)];
nsmall=size(small,1);
nnurse=size(nurse,1);
nfat=size(fat,1);
smallsale=small(find(small(:,6)==2),:);%可卖的副产品猪
nursesale=nurse(find(nurse(:,6)==2),:);
fatsale=fat(find(fat(:,6)==2),:);

%% 要卖的猪的数量
toln=min(size(A,1)-limit,naver);
nsalen=ceil(toln/(nnurse+nfat)*nnurse);
nsalef=toln-nsalen;
nsales=nsmall-limits;
% nsales=0;
%% 挑选要卖的猪仔
ss1=smallsale(find(smallsale(:,3)<=28),:);
ss1=sortrows(ss1,-3);

```

```

ss=[];
% for i=1:nsales
%     ss=[ss;ss1(i,:)];
% end
if nsales~=0
    ss=selectpigs(smallsale,nsales);
end
% nsales=0;

%% 挑选要卖的保育猪
sn1=nursesale(find(nursesale(:,3)<=56),:);
sn1=sortrows(sn1,-3);
sn=[];
% for i=1:nsalen
%     sn=[sn;sn1(i,:)];
% end
if nsalen~=0
    sn=selectpig(nursesale,nsalen);
end

%% 挑选要卖的育肥猪
% sf1=fatsale(find(fatsale(:,3)>=76),:);
sf1=fatsale;
sf1=sortrows(sf1,-3);
sf=[];
for i=1:nsalef
    sf=[sf;sf1(i,:)];
end

%% 所有要卖的猪状态更新
totalsale=[ss;sn;sf];%仔猪，保育，育肥
Anew=A;
for i=1:size(totalsale,1)
    Anew(find(Anew(:,1)==totalsale(i,1)),:)=[];
end
for i=1:size(Anew,1)
    Anew(i,1)=i;
end

%% 本周销售计划输出
weight1=0;
count1=0;
weight2=0;
count2=0;
weight3=0;
count3=0;
weight4=0;
count4=0;
for i=1:size(totalsale,1)
    dweight=dw(totalsale(i,3));

```

```

if dweight<=20
    weight1=weight1+dweight;
    count1=count1+1;
elseif dweight>=25&dweight<=50
    weight2=weight2+dweight;
    count2=count2+1;
elseif dweight>=50&dweight<=100
    weight3=weight3+dweight;
    count3=count3+1;
elseif dweight>=100
    weight4=weight4+dweight;
    count4=count4+1;
end
end
count=count1+count2+count3+count4;%数据的简单核算
saleplan=[weight1,weight2,weight3,weight4];

end

```