



2019 年湖南省高校第四届研究生数学建模竞赛参赛承书

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

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）： 国防科技大学

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

题目：基于粒子群优化与仿真的猪场连续生产销售策略研究

摘要：

在生猪养殖行业中，高品质的大量猪肉需求要求我国生猪肉饲养方式向专业化、集约化和规模化饲养方式转变。针对一个具体的养猪场连续生产问题，本文根据各品种市价、场内运载能力、产能约束等条件，合理配置每周农场配种计划及副产品销售计划，以实现连续生产的收益最大化。

首先，在进行数据处理和问题分析后，本文将该问题建立为一个可求解的混合整数规划模型。考虑对已知数据进行整理和拟合，我们得到了猪的日龄与饲喂成本，重量和销售价格的函数模型。基于此，本文创造性地将各个年龄段的猪的数量建模为一组决策变量。在找到各个变量间的内在联系后，本文将原问题拆分为五个事件（母猪分娩、猪死亡、种猪出售、副产品出售、生产猪替换）、三个约束（场地约束，运输约束，存栏约束）和一个目标（最大化利益），使原问题得到简化，并据此建立了相应的子模型、约束条件和目标函数。

其次，为实现养猪场收益最大化，我们将养猪场的经营策略分为两步“如何养”和“怎么卖”两个部分，分别为配种计划的选择和销售计划的制定。总体求解流程为，先利用启发式算法制定初始配种策略和销售策略，然后采用仿真思想分析得到销售最大化策略以优化销售计划，利用粒子群优化算法以配种策略中的参数，使得模型在两种计划的配合下实现目标函数最大化（见附件 7，8），并在仿真过程中根据各个阶段的猪的数量情况，计算得到饲料消耗（见附件 9）。基于此，设计了对比试验对销售收益最大化策略进行验证，实验结果表明，在随机策略下养猪场全年累计利益为 988.78 万元，而采用销售利益最大化策略生成销售计划的年利益达 1162.23 万元（增长 14.92%）。最后，利用粒子群算法优化配种策略参数，所生成的最优配种计划仿真得到年利益大 1239.91 万元（增长 20.25）。

关键词： 生猪经营管理、混合整数规划模型、粒子群算法、

一、问题背景

某养猪场现有各类猪只共约 1 万头，希望在满足猪只生产规律、猪舍容量、运输能力等约束条件下，制定连续生产、销售计划，包括配种、产仔、销售、更新等环节，以实现充分利用现有资源，利润最大化。养猪场中一共有 A, B, C, D, E 和 F 类猪，其配种信息如下：

- A 系公猪与 B 系母猪配种，产出 F 系母猪为种猪产品，F 系公猪全部作为副产品销售。
- C 系公猪与 D 系母猪配种，产出 E 系公猪为种猪产品，E 系母猪全部作为副产品销售。
- 纯系配种可产出用于自留替换的种猪。如 B 系公猪与 B 系母猪配种，产出 B 系母猪用于生产母猪的更新替换，B 系公猪可以作为副产品出售，其他类似。

1. 猪只生产全过程包括配种怀孕、分娩哺乳、保育、育肥、更新等环节。由于题目中信息不够精确，我们结合附件 6 中的附件存栏，通过数据的统计和分析，得到精确到天的生长状态转换时间，如下表所示：

表 1 生长状态以及死亡率和猪日龄的对应关系

	日龄（单位：天）	备注
哺乳期	0-27	存活率 90%
保育期	28-76	存活率 97%
育肥期	77-长期	存活率 98%
F 系母猪选种	105	选种率 65%
E 系公猪选种	161	选种率 40%
母猪配种	210-长期	至少 1100 头
公猪取精	301-长期	至少 28 头
母猪怀孕	时长 116 天	分娩率 88%
生产公猪替换	666	替换率为 65%
生产母猪替换	940	替换率为 100%

此外，猪的补充生长状态信息如下：

- 母猪年产窝数约 2.45。
 - 母猪一窝产活仔数：F 系母仔猪约 11-12 头，E 系公仔猪约 9-9.5 头，纯系仔猪约 10 头（附件 1），仔猪公母比例 1:1。
2. 该猪场现有各类猪舍 36 间，其猪舍分配情况如下：
- 8 间产房，每间产房 24 个产床（即最多可同时养 24 头猪）。每头母猪在产房里待 24 天。因产房冲洗、干燥、维修、消毒等操作约需 5 天周转，故可按 6 间产房计算容量，即产房内正常母猪存栏总数不超过 24*6 头。
 - 14 间保育舍，每间 12 个栏位，每个栏位最多养 6-22.7kg 的猪只约 24 头，或者 22.7-25kg 的猪只约 18 头。养至约 25kg 后转育肥舍。
 - 14 间育肥舍，每间 24 个栏位，每个栏最多养 22.7-34kg 的猪只约 46 头，或者 34-120kg 的猪只约 23 头。

3. 该猪场的主要产品为种猪，不能作为种猪出售的猪可以作为副产品销售。

- 副产品猪按照单个猪的种类分为四类：

表 2 单个猪重量与副产品猪种类对应关系

副产品猪种类	单个猪重量
副产品一	0~20KG
副产品二	25KG~50KG
副产品三	50KG~100KG
副产品四	100KG~120KG

- 副产品销售时需使用场内运输车对接，除去消毒、隔离时间，每辆运输车一周最多使用3次，不同体重、品种的猪可以混装。

4. 为保证连续生产，避免资源浪费，通常要求猪只存栏总数不低于 9000 头，其中生产母猪不低于 1100 头，生产公猪不低于 28 头。种猪销售价格全年稳定，生产母猪单价 1600 元/头，生产公猪单价 1250 元/头。通常种公猪销售体重不超过 114kg，种母猪销售体重不超过 105kg。

假设当前时间为 2019 年 1 月 1 日，该猪场现有 A、B、C、D、E、F 等 6 个品系的种猪和副产品猪存栏共约 1 万头。猪场根据订单信息及市场需要的变化情况，预测出 2019 年各品系种猪月销售计划。根据各品种市价、场内运载能力、产能约束等条件，合理配置每周农场配种计划（含配种猪只信息，总头数等）及副产品销售时间及品种，实现连续生产的收益最大化，请完成包括但不限于如下具体任务：

- (1) 制定每周配种计划，及生产公猪、生产母猪的更新计划，填写附件 8（说明：全年按 52 周计。配种计划可按天制定，再转换为周计划）；
- (2) 制定四种副产品每周的销售计划，填写附件 9；
- (3) 测算每周各类饲料消耗量，填写附件 10。

二、数据处理

由于问题给出的数据较为复杂繁琐，我们首先对其中数据进行简化和拟合，得到适合建模的函数模型。

(一)根据附件 1 对猪日龄与饲喂成本进行拟合，得到拟合图像如下：

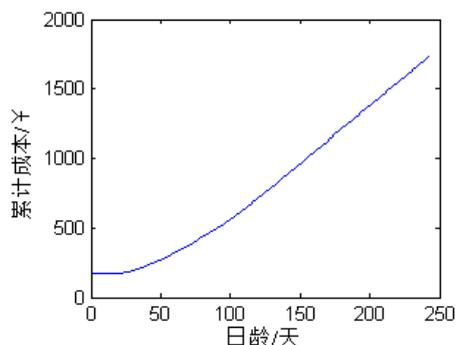


图 1 日龄-饲料关系图

依据附件 1 提供的数据，可以得到猪的日龄 $x(x \in R^{++})$ 与饲喂成本之间的多项式拟合函数如下：

$$\omega(x) = \begin{cases} -0.0008x^3 + 0.0443x^2 + 0.7746x + 137.64 & x \geq 20 \\ 171 & x < 20 \end{cases} \quad (1)$$

(二) 根据附件 2 生长数据，我们发现在日龄为 120 天之后的重量数据较为分散，无法进行直接拟合，因此，我们充分考虑到问题的特点，发现同样日龄下，育肥猪和种猪的体重具有较大差异。基于此，我们对数据集进行了聚类处理，并分别对其进行拟合。

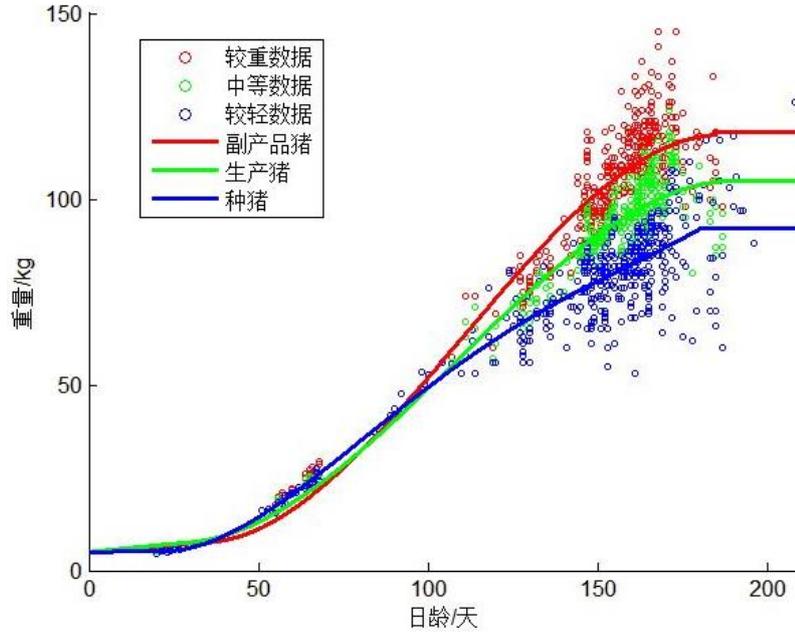


图 2 日龄-重量关系图

然后，我们得到不同种类猪的日龄 $x(x \in R^{++})$ 与猪的重量 $weight$ 之间的多项式拟合函数如下：

$$Weight_u(x) = \begin{cases} 0.078x + 5 & x < 33 \\ -6.144 \times 10^{-5}x^3 + 0.02029x^2 - 1.157x + 25.87 & 33 \leq x < 185 \\ 0.001x + 117.815 & x \geq 185 \end{cases} \quad (2)$$

$$Weight_p(x) = \begin{cases} 0.088x + 5 & x < 29 \\ -4.416 \times 10^{-5}x^3 + 0.01437x^2 - 0.6676x + 15.9 & 29 \leq x < 189 \\ 0.001x + 104.77 & x \geq 189 \end{cases} \quad (3)$$

$$Weight_x(x) = \begin{cases} 0.015x + 5 & x < 20 \\ 2.98 \times 10^{-7}x^4 - 1.44 \times 10^{-4}x^3 + 0.023x^2 - 0.83x + 13.77 & 20 \leq x < 180 \\ 0.001x + 91.93 & x \geq 180 \end{cases} \quad (4)$$

(三) 依据附件 4-副产品售价预测，可以得到不同的副产品猪在不同时期 t 的销售价格，即不同重量的猪的价格 $Price(weight, t)$ 。又已知猪的日龄与重量之间

具有 $Weight_u(x)$ 的一一对应关系，即已知一头猪的日龄，可以大致推测其重量 $weight=Weight(x)$ ；综上，我们可以得到单个日龄为 x 的副产品猪在 t 时期的销售价格为：

$$Earn(x,t) = Price(Weight_u(x),t) \cdot Weight_u(x) \quad (5)$$

其中， $x \in Z, t \in Z^{++}, x < t$ ，其函数示意图如下：

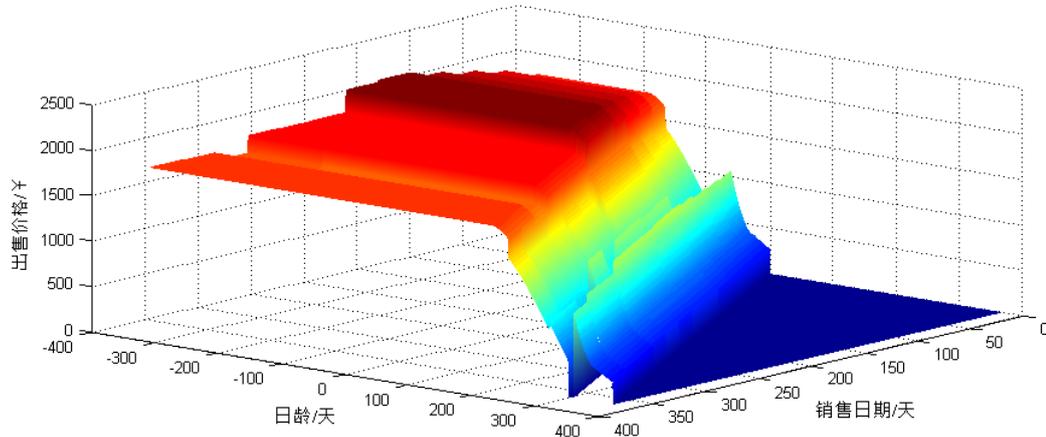


图3 $Earn(x,t)$ 函数图像

三、问题基本假设

1. 不考虑瘟疫等小概率事件；
2. 附件6-猪只存栏信息中，假设其中部分生产母猪处于已配种状态；
3. 在满足车辆约束的前提下，所有猪作为副产品销售时能全部卖出，不考虑市场饱和的情况；

四、问题分析与数学建模

通过对数据的处理，我们得到了三组函数关系式，分别为猪的日龄与其饲料成本，猪的日龄与其销售价格以及猪的日龄与其重量变化，为后续建模和问题求解奠定基础。

4.1 问题分析

本文的目标是依据订单信息与市场各品种市价、场内运载能力、产能约束等条件，合理配置每周农场配种计划（含配种猪只信息，总头数等）及副产品销售时间及品种，实现**连续生产的收益最大化**。现依据问题背景和已知假设条件，我们可以总结出影响各个种类的猪的数量变化的事件如下：

(1) 母猪分娩产下仔猪：

其中A系公猪与B系母猪配种，产出F系仔猪；C系公猪与D系母猪配种，产出E系仔猪；纯系配种可产出用于自留替换的仔猪；E系和F系不参与养猪场配种。

(2) 按种猪月销售计划出售种猪:

当种猪成长到待售种猪阶段时,即可按月销售计划选择待售种猪中销售利益最大的种猪进行出售。通常种公猪销售体重不超过114kg,种母猪销售体重不超过105kg。

(3) 在满足运输约束前提下,销售副产品猪。

副产品一销售体重约 $\leq 20\text{kg}$,副产品二约25-50kg(20kg-25kg的猪不能销售),副产品三约50-100kg,副产品四约100kg-120kg。

(4) 生产母猪的替换与生产公猪的替换。

对于生产母猪而言,从第一次配种起,2年内可不更新,使用2年以上可考虑更新替换,通常每年更新替换率约65%。对于生产公猪,每年更新替换率通常为100%。更新替换掉的生产猪,可以作为副产品出售。

(5) 从仔猪到待出售猪之间的正常损耗与死亡。具体可见表(?)的备注栏。其次,依据对问题描述的进一步分析,我们可以归纳出该问题的约束条件如下:

(1) 养猪场场地约束

➤ 产房约束: 8间产房,每间产房24个产床(即最多可同时养24头猪)。每头母猪在产房里待24天。因产房冲洗、干燥、维修、消毒等操作约需5天周转,故可按6间产房计算容量,即产房内正常母猪存栏总数不超过 24×6 头。

➤ 保育舍约束: 14间保育舍,每间12个栏位,每个栏位最多养6-22.7kg的猪只约24头,或者22.7-25kg的猪只约18头。养至约25kg后转育肥舍。

➤ 育肥舍约束: 14间育肥舍,每间24个栏位,每个栏最多养22.7-34kg的猪只约46头,或者34-120kg的猪只约23头。

(2) 销售副产品时,运输车能力约束:

副产品销售时需使用场内运输车对接,除去消毒、隔离时间,每辆运输车一周最多使用3次,不同体重、品种的猪可以混装。

(3) 可持续发展约束: 为保证连续生产,避免资源浪费,需要对猪场的存栏数做出限制:

- 猪只存栏总数不低于9000头
- 生产母猪不低于1100头
- 生产公猪不低于28头

4.2 数学建模

通过对问题的进一步分析,我们将该复杂问题拆分为三个方面理解,分别是目标,约束条件和影响猪内部数量的事件。目标是为了实现养猪场利益最大化;约束条件包括厂房约束,运输约束和可持续发展约束;事件包括母猪分娩,仔猪生长过程中的死亡,出售种猪和副产品猪以及生产猪的更新替换。

4.2.1 决策变量

假设处于生长各个日龄的种猪数量为 $x(t) = \{x_1(t), x_2(t), \dots, x_k(t)\}$,其中 $x_1(t)$ 表示第 t 天(从2019年1月1日起)日龄为1的种猪数量, $x_2(t)$ 表示第 t 天日龄为2的种猪数量, $x_k(t)$ 表示第 t 天(从2019年1月1日起)日龄为 k 的种猪数量,因此,我们不难得到:

$$x_i(t+1) = x_{i-1}(t) \quad i, t \in Z \quad (6)$$

假设处于生长各个日龄的副产品猪数量为 $u(t) = \{u_1(t), u_2(t), \dots, u_c(t)\}$,其中

$u_c(t)$ 表示第 t 天（从2019年1月1日起）日龄为 c 的种猪数量，因此，我们不难得到：

$$u_i(t+1) = u_{i-1}(t) \quad i, t \in Z \quad (7)$$

假设处于各个日龄的怀孕母猪数量为 $y(t) = \{y_1(t), y_2(t), \dots, y_c(t)\}$ ，其中 $y_i(t)$ 表示第 t 天（从2019年1月1日起）怀孕天数为 c 的母猪数量，因此可以得到 $y(t)$ 的迭代公式如下：

$$y_i(t+1) = y_{i-1}(t) \quad i, t \in Z \quad (8)$$

假设处于各个日龄的生产公猪和生产母猪（包括后备母猪）数量为 $p^s(t) = \{p_k^f(t), p_l^m(t), p_{k+1}^f(t), p_{l+1}^m(t), \dots, p_{k+N_f}^f(t), p_{l+N_m}^m(t)\}$ ，其中 $p_k^f(t)$ 表示第 t 天（从2019年1月1日起）日龄为 k 的生产母猪数量， $p_l^m(t)$ 表示第 t 天日龄为 l 的生产公猪数量， $k + N_f$ 和 $l + N_m$ 分别表示生产母猪和生产公猪的最大日龄，因此可以得到的迭代公式如下：

$$p_k^f(t+1) = p_{k+1}^f(t) \quad k, t \in Z \quad f \text{ represent female} \quad (9)$$

$$p_l^m(t+1) = p_{l+1}^m(t) \quad k, t \in Z \quad m \text{ represent male} \quad (10)$$

4.2.2 事件建模

（一）子模型 1：猪死亡损耗

由题意，仔猪数量在生长过程中会面临一些死亡损耗，我们对此种数量变化进行了事件建模如下：

$$\sum_{i=28}^{76} x_i(t) = \alpha \sum_{j=1}^{27} x_j(t) \quad i, t \in Z, \quad \alpha \in [0, 1] \quad (11)$$

$$\sum_{i=77}^k x_i(t) = \beta \sum_{j=28}^{76} x_j(t) \quad i, t \in Z, \quad \beta \in [0, 1] \quad (12)$$

其中 $\sum_{i=1}^{28} x_i(t)$ 代表处于哺乳期的种猪数量， $\sum_{i=28}^{76} x_i(t)$ 代表处于保育期的种猪数量， α 代表哺乳期种猪的存留率，公式(11)代表种猪从哺乳期到保育期的数量变化；同理，公式(12)代表种猪从保育期到待售期的数量变化。

$$\sum_{i=28}^{76} u_i(t) = \alpha \sum_{j=1}^{27} u_j(t) \quad i, t \in Z, \quad \alpha \in [0, 1] \quad (13)$$

$$\sum_{i=77}^K u_i(t) = \beta \sum_{j=28}^{76} u_j(t) \quad i, t \in Z, \quad \beta \in [0, 1] \quad (14)$$

其中 $\sum_{i=1}^{28} u_i(t)$ 代表处于哺乳期的副产品猪数量， $\sum_{i=28}^{76} u_i(t)$ 代表处于保育期的副产品猪数量， α 代表哺乳期副产品猪的存留率，公式(13)代表副产品猪从哺乳期到保育期的数量变化；同理，公式(14)代表副产品猪从保育期到待售期的数量变化。

(二) 子模型2: 出售种猪

如果在第 t 天时出售种猪 $L(t)$ 只, 即可以用下列式子表示:

$$\exists x_m(t) \in x(t) \quad (15)$$

$$s.t. \quad \sum_{i=m}^k x_i(t) = L_1(t) \quad (16)$$

$$\sum_{i=m-1}^k x_i(t) = L_2(t) \quad (17)$$

$$L_1(t) \leq L(t) \leq L_2(t) \quad (18)$$

公式(15)~(17)表示从日龄最大的种猪开始出售, 一直出售到日龄为 $(m-1)$ 的种猪。所以, 经过一次售出事件后, 可以得到 $x(t)$ 的更新公式如下:

$$x'_{m-1}(t) = x_{m-1}(t) + (L(t) - \sum_{i=m}^k x_i(t)) \quad (19)$$

$$x(t) = \{x_1(t), x_2(t), \dots, x'_{m-1}(t)\} \quad (20)$$

(三)子模型3: 出售副产品猪

如果在第 t 天时出售副产品 $C(C=1,2,3,4)$ 一共 $W(t)^{(C)}$ kg, 即可以用下列式子表示:

$$\exists u_m(t)^{(C)} \in u(t)^{(C)} \quad (21)$$

$$s.t. \quad \sum_{i=m_C}^{k_C} u_i(t) Weight_u(x) = W_1(t)^{(C)} \quad (22)$$

$$\sum_{i=m_C-1}^{k_C} u_i(t) Weight_u(i) = W_2(t)^{(C)} \quad (23)$$

$$W_1(t)^{(C)} \leq W(t)^{(C)} \leq W_2(t)^{(C)} \quad (24)$$

公式(21)~(24)表示将重量转化为日龄后, 同样从最大日龄开始出售副产品猪。其中, k_c 表示 $C(C=1,2,3,4)$ 类副产品猪的重量分界线对应猪的日龄, 即

$$k_1 = [Weight^{-1}(20)] \quad k_2 = [Weight^{-1}(25)] \quad (25)$$

$$k_3 = [Weight^{-1}(50)] \quad k_4 = [Weight^{-1}(100)] \quad (26)$$

所以, 经过一次副产品售出事件后, 可以得到 $u(t)$ 的更新公式如下:

$$u'_{m_C-1}(t) = u_{m_C-1}(t) + \left[\frac{W(t)^{(C)} - \sum_{i=m_C}^{k_C} u_i(t) Weight_u(i)}{Weight_u(m_C - 1)} \right] \quad (27)$$

$$u(t)^{(1)} = \{u_1(t), u_2(t), \dots, u_{m_1-1}(t)\} \quad u(t)^{(2)} = \{u_{k_2}(t), u_2(t), \dots, u_{m_2-1}(t)\} \quad (28)$$

$$u(t)^{(3)} = \{u_{k_3}(t), u_2(t), \dots, u_{m_3-1}(t)\} \quad u(t)^{(4)} = \{u_{k_4}(t), u_2(t), \dots, u_{m_4-1}(t)\} \quad (29)$$

$$u(t) = \bigcup_{C=1,2,3,4} u(t)^{(C)} \quad (30)$$

其中，公式(27)代表第 C 类副产品销售后，日龄为 $(m_C - 1)$ 之后的猪（包括一部分日龄为 $(m_C - 1)$ ）全部被当作副产品销售后，日龄为 $(m_C - 1)$ 的猪的剩余数量。公式(28)~(30)表示所有类副产品在一次销售完成后，整个副产品猪的在各个日龄阶段的剩余数量。

（四）子模型 4：母猪生产

由题意，母猪生产涉及到不同种类之间的数量变化，其变化符合遗传学规律，比如题中明确其中 A 系公猪与 B 系母猪配种，产出 F 系仔猪。因为配种信息是已知的，所以我们可以得知临产母猪将产下何种仔猪。此时我们规定在第 t 天将产下 X 系仔猪的母猪数量记为 $y_{116}^X(t)$ ，易得：

$$y_{116}^X(t) \leq y_{116}(t) \quad (31)$$

$$y^X(t) \cdot \delta_X = u^X(t+1) + x^X(t+1) \quad (32)$$

其中， $y_{116}(t)$ 表示在第 t 天所有临产母猪数量， $u^X(t+1)$ 表示在第 $(t+1)$ 天日龄为 1 的 X 系种猪数量， $x^X(t+1)$ 表示在第 $(t+1)$ 天日龄为 1 的 X 系副产品猪数量。

（五）子模型 5：生产母猪\生产公猪替换

由题意，对于生产母猪而言，从第一次配种(日龄为 301)使用 2 年以上可考虑更新替换，通常每年更新替换率约 65%。对于生产公猪（日龄为 210），每年更新替换率通常为 100%，即

$$\forall j \geq (301 + 365) \quad p_j^m(t) = 0 \quad (33)$$

$$\forall j \geq (210 + 365 \times 2) \quad p_j^m(t)' = 0.65p_j^m(t) \quad (34)$$

4.2.3 目标函数

由题意，我们需要设置合适的配种计划和销售计划以实现猪场连续生产的利益最大化。在上述问题分析和模型中，猪场的副产品猪销售计划可以表示为

$$W_C(t) \quad s.t. \quad C = 1, 2, 3, 4 \quad (35)$$

而配种计划可以表示为设置

$$y_1^X(t) \quad s.t. \quad X = A, B, C, D, E, F \quad (36)$$

首先我们计算按照销售计划出售的副产品猪的总收益，在上一章的数据处理中，我们得到公式(5)，即一头副产品猪的价格可以由其日龄和销售日期确定。依据副产品的销售计划，我们可以得到副产品年总收入 F_{income}

$$F_{income} = \sum_{t=1}^{365} \sum_{C=1}^4 W_C(t) \cdot Price(t, k_C) \quad (37)$$

其次，由于种猪的销售计划已经确定，并且种猪的价格全年稳定，其售卖种猪年总收入可以用常量 Z_{income} 表示。

再次，在由公式(38)可以得到猪的日龄与饲喂成本之间的函数，此时可得到一年内的饲喂总成本为：

$$B_{cost} = \sum_{t=1}^{365} \left(\sum_i u_i(t) \alpha(i) + \sum_j x_j(t) \alpha(j) + \sum_k p_k^f(t) \alpha(k) + \sum_l p_l^m(t) \alpha(l) + \sum_a y_a(t) \alpha(a) \right) \quad (38)$$

最后，考虑车辆运输成本 T_{cost} ，设全年使用运输车一 p 次，使用运输车二 q 次，则由附件 5 可以得到：

$$p = \sum_t (0.5 \operatorname{sgn}(\sum_C W_C(t) - MC_1) - 0.5) \quad (39)$$

$$q = \sum_t (0.5 \operatorname{sgn}(MC_2 - \sum_C W_C(t)) - 0.5) \quad (40)$$

其中， MC_1 和 MC_2 分别为运输车一和运输车二的单次最大运载能力。已知运输车一盒运输车二的单次运载价格分别为 $P_{cost}^{(1)}$ 和 $P_{cost}^{(2)}$ ，可得运输车总成本为：

$$P_{cost} = P_{cost}^{(1)} \cdot p + P_{cost}^{(2)} \cdot q \quad (41)$$

综上，我们可得目标函数为

$$\text{Max Profit} = F_{income} + Z_{income} - B_{cost} - P_{cost} \quad (42)$$

4.2.4 约束条件

(一) 养猪场场地约束

- 产房约束

$$\forall t \in [1, 365], y_{116}(t) \leq S_{cf} \quad (43)$$

其中 S_{cf} 为产房能容纳的最多母猪数量，取值为 24×6

- 保育室约束

已知在保育室中猪的重量处于 Wb_1 到 Wb_2 之间的猪的数量为

$\sum_{\hat{i}=Weight_x^{-1}(Wb_1)}^{Weight_x^{-1}(Wb_2)} x_i(t) + \sum_{\hat{i}=Weight_u^{-1}(Wb_1)}^{Weight_u^{-1}(Wb_2)} u_i(t)$ ，处于 Wb_3 到 Wb_4 之间的保育猪数量为

$\sum_{\hat{i}=Weight_x^{-1}(Wb_3)}^{Weight_x^{-1}(Wb_4)} x_i(t) + \sum_{\hat{i}=Weight_u^{-1}(Wb_3)}^{Weight_u^{-1}(Wb_4)} u_i(t)$ ，则保育室约束可以表示为：

$$\frac{\sum_{\hat{i}=Weight_x^{-1}(Wb_1)}^{Weight_x^{-1}(Wb_2)} x_i(t) + \sum_{\hat{i}=Weight_u^{-1}(Wb_1)}^{Weight_u^{-1}(Wb_2)} u_i(t)}{S_{BY1}} + \frac{\sum_{\hat{i}=Weight_x^{-1}(Wb_3)}^{Weight_x^{-1}(Wb_4)} x_i(t) + \sum_{\hat{i}=Weight_u^{-1}(Wb_3)}^{Weight_u^{-1}(Wb_4)} u_i(t)}{S_{BY2}} \leq S_{BY} \quad (44)$$

其中， S_{BY} ， S_{BY1} 和 S_{BY2} 为题目已知参数，分别为 14×12 ，24，18。

- 育肥室约束

已知在育肥室中猪的重量处于 Wy_1 到 Wy_2 之间的猪的数量为

$$sy_1 = \sum_{\hat{i}=Weight_x^{-1}(Wy_1)}^{Weight_x^{-1}(Wy_2)} x_i(t) + \sum_{\hat{i}=Weight_u^{-1}(Wy_1)}^{Weight_u^{-1}(Wy_2)} u_i(t) + \sum_{\hat{i}=Weight_p^{-1}(Wy_1)}^{Weight_p^{-1}(Wy_2)} \left(\sum_s p_i^s(t) + y_i(t) \right) \quad (45)$$

处于 Wb_3 到 Wb_4 之间的猪数量为

$$sy_2 = \sum_{i=Weight_x^{-1}(Wy_3)}^{Weight_x^{-1}(Wy_4)} x_i(t) + \sum_{i=Weight_u^{-1}(Wy_3)}^{Weight_u^{-1}(Wy_4)} u_i(t) + \sum_{i=Weight_p^{-1}(Wy_3)}^{Weight_p^{-1}(Wy_4)} (\sum_s p_i^s(t) + y_i(t)) \quad (46)$$

则育肥室约束可以表示为:

$$\frac{sy_1}{S_{YF1}} + \frac{sy_2}{S_{YF2}} \leq S_{YF} \quad (47)$$

其中, S_{YF} , S_{YF1} 和 S_{YF2} 为题目已知参数, 分别为 $14*24$, 46 , 23 。

(二) 运输能力约束

根据题意, 运输副产品的运输车每周最多运送三次, 则可由下式表示:

$$\forall l \in [1, 365], l \in Z \quad \sum_{t=7l-6}^{7l} \text{sgn}(\sum_{C=1}^4 W_C(t)) \leq 3 \quad (48)$$

(三) 可持续发展约束

- 存栏总数不低于 9000 头

$$\forall t \in [1, 365], \sum_i x_i(t) + \sum_j u_j(t) + \sum_k y_k(t) + \sum_a \sum_s p_a^s(t) \leq 9000 \quad (49)$$

- 生产母猪不低于 1110 头

$$\forall t \in [1, 365], \sum_a p_a^f(t) + \sum_k y_k(t) \geq 1110 \quad (50)$$

- 生产公猪不低于 30 头

$$\forall t \in [1, 365], \sum_a p_a^m(t) \geq 30 \quad (51)$$

五、问题求解

本题的目标是制定合适的经营策略, 使得养猪场利益实现最大化。我们将经营策略分为“如何养”和“如何卖”两个部分。具体而言, 就是在满足模型中各类约束的前提下, 制定合适的配种计划和销售策略。

求解该问题需要对两个计划进行优化, 如果同时进行优化, 那么解空间过于庞大, 难以求得最优解。因此, 我们的总体求解流程为, 先利用启发式算法制定配种策略和销售策略。利用粒子群优化算法对配种策略中的参数进行优化, 并且采用销售收益最大化策略来优化销售计划, 使得在两种计划的配合下, 能够产生最大收益。并在仿真过程中根据各个阶段的猪的数量情况, 计算得到饲料消耗。

5.1 初始配种计划

本节主要工作按照给定的种猪销售计划, 给出初始配种策略。

5.1.1 问题补充假设

1、由于题中没有给出目前已有母猪的怀孕信息，根据生产均匀连贯性原则，考虑充分利用产房，假设每天有 6 窝猪仔出生。

2、由于怀孕时间为 116 天，则前 116 天生产的猪仔种类不是根据本文制定的配种计划产生。因此假设各类仔猪按照一定的概率出生。其中 A、B、C、D、E、F 的概率依次设定为[0.1,0.1,0.1,0.1,0.3,0.3]，此设定为初始设定，后续将会用粒子群优化算法对其进行优化。

5.1.2 求解原则：

第一步：更新猪群的信息，包括猪的日龄、状态、类别、重量、生产母猪的生产日、生产母猪的生产种类、出售日期。

第二步：根据 ABCDEF 类销售种猪在 N 天之后的出售需求，进行配种，其中 N 包括猪的怀孕时间和成长时间，不同类型的猪时间不同。当天的 6 次配种机会还有剩余的话，就进行随机配种，如果没有则进入下一步。

第三步：如果时间戳小于 116 天，则按概率生产 6 窝猪仔并确定种猪，否则进入下一步。

第四步：根据题设要求，进行副产品和种猪的销售，并计算收益。进入下一步。

第五步：根据目前存栏量，计算日饲料消耗量。判断时间是否小于 364 天，如果是，则结束，否则更新时间戳，跳入第一步。

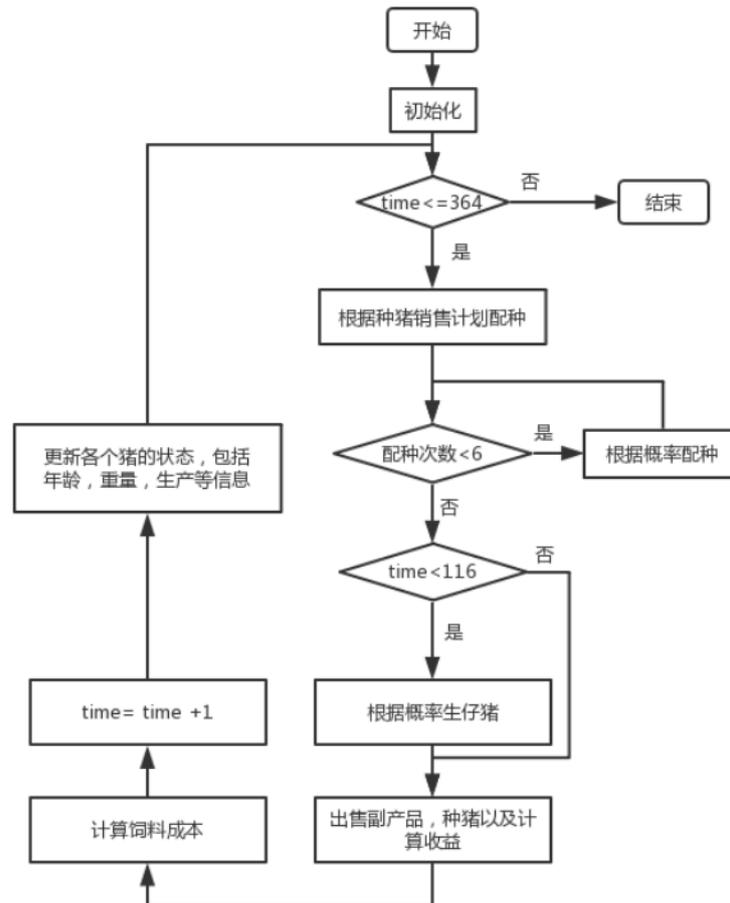


图 4. 配种计划制定流程图

5.2 销售计划制定

本节在接收到上一节给出的初始配种计划后,分析并计算模型中的各个约束条件,以得到求解前提和结论。其次,对相应数据进行分析之后,可以得到利益最大化的启发式规则,并据此对销售计划进行迭代优化。

5.2.1 问题补充假设

1. 由于种猪销售是按照头数计算价格而不是体重来计算,因此,为了实现饲养成本最小化,假设在每个月的第一天完成种猪的销售。

5.2.2 销售问题分析

销售计划主要包括每月种猪的销售以及每周副产品猪的销售。首先考虑种猪的销售。按照假设,我们制定种猪销售策略为:当月初种猪数量满足当月销售计划时,在月初对种猪进行售卖,否则依次后推种猪售卖时间,以节约饲料成本。

下面进行副产品猪销售计划的制定,根据运输车辆的约束,我们以周为单位制定销售计划。在副产品猪销售时,需要考虑的约束有猪场厂房约束、可持续发展约束、运输车辆约束。

(1) 猪舍容量约束

按照题目所给信息,共有三种猪舍,分别为产房、保育舍和育肥舍,每种猪舍都有各自的产能约束。其中产房只对于母猪的存栏总数进行限制,即生产母猪数量不超过 24×6 头,而还在哺乳期的仔猪的数量并无限制,仔猪数量记为 N_1 。

按照体重可将保育舍中的猪分为两种,体重为 6-22.7kg 的猪的总数记为 N_2 ,体重为 22.7-25kg 的猪的总数记为 N_3 ,按照保育舍产能约束,两种类型的猪的数量约束见公式(44)。育肥舍中的猪按照体重也可分为两种,体重为 22.7-34kg 的猪的总数记为 N_4 ,体重为 34-120kg 的猪的总数记为 N_5 ,按照育肥舍产能约束,两种类型的猪的数量约束见公式(47)。

依据 5.1 节制定的配种计划,每天有 6 头母猪生产,按照最大生产量,每天有 21×6 只仔猪出生。若不考虑售卖种猪和副产品,时间向前推进 100 天,探究各猪舍的猪数量的演化情况,如图 5 所示:

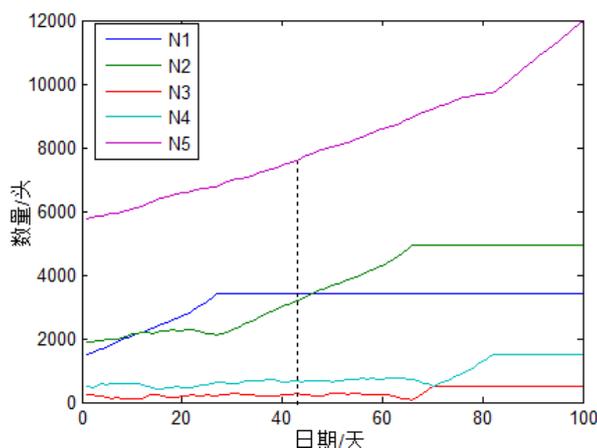


图 5. 未销售情况下各类猪数量的演化

从图中可以看出，猪的总数一直在增加，当达到 42 天时，猪的数量超过了猪舍的承载能力。此外，可以看出 N_3 和 N_4 的数量一直比较稳定，我们计算 N_3 和 N_4 前 42 天的平均值其结果分别为 491 和 280。可以假设，在未超过猪舍承载能力的情况下，这两类猪的数量保持稳定在平均值，则通过式 (44) 和式 (47) 可以计算出 N_2 和 N_5 的最大值，即 $N_2^{\max} = 3377, N_5^{\max} = 7588$ 。为了维持猪的种群结构的稳定，假设在销售时只考虑减小 N_2, N_5 的数量，即只卖体重在 6-22.7kg 和 34-120kg 的猪，控制销售数量使得 N_2, N_5 一直小于最大上限，即可保持种群结构稳定和满足猪舍容量约束。并且这两种体重的猪也可以满足四类副产品的分类。

(2) 猪只存栏总数约束

该猪场为保证连续生产，要求猪只存栏总数不低于 9000 头，因此某一周销售猪只数量存在上限 S_1 。为保证种群数量的稳定，需要尽量使得每周入群的猪只数量等于每周出群的猪只数量，即某一周销售猪只数量 S 尽量等于该周出生仔猪的数量 S_2 。综合考虑，即可得出某一周销售猪只的数量应为 $S = \min(S_1, S_2)$ 。

考虑到只卖两种体重类型的猪，需要将一周的销售猪只数量按照一定规则分配给两种体重的猪群。假设体重在 6-22.7kg 的猪只现有 N_2 头，其销售数量为 K_1 ，体重在 34-120kg 的猪只现有 N_5 头，其销售数量为 K_2 ，则需要按照两类猪群的富余量进行销售总数 S 的划分，即

$$K_1 = \frac{N_5^{\max} - N_5}{N_5^{\max}} \cdot S \left/ \left(\frac{N_2^{\max} - N_2}{N_2^{\max}} + \frac{N_5^{\max} - N_5}{N_5^{\max}} \right) \right., K_2 = S - K_1 \quad (52)$$

(3) 运输车辆约束

该猪场共有两辆运输车，每周可以分别调用三次，即单周最大运载重量为 $1.341 \times 10^5 \text{kg}$ 。假设每天有 6 只生产母猪以最大生产量进行生产，即每周最多出生 $21 \times 6 \times 7$ 只仔猪。因此每周同样需要卖出 $21 \times 6 \times 7 = 882$ 只猪。根据日龄与体重之间的变化关系可知，单头猪实际体重不超过 120kg，所以每周最大销售重量不超过 $1.0584 \times 10^5 \text{kg}$ ，远小于单周最大运载重量。因此，在本文制定的销售策略中，运输车辆的负载不构成约束。

5.2.2 销售收益最大化

本节考虑在每周的销售中，都实现该周销售收益的最大化。假设某一周的销售总量已经定为 S ，如何在现有猪群中选择 S 头猪，成为销售收益最大化的关键。由于每月各类副产品价格的变化、猪只的体重变化以及饲喂成本的变化，对于每只猪都有一个最佳销售日期。以 2019 年 1 月 1 日为准，在该日之前出生的猪日龄记为负数，在该天之后出生的猪日龄记为正数，则可以计算出日龄与最佳销售日期之间的关系，如图 6 所示：

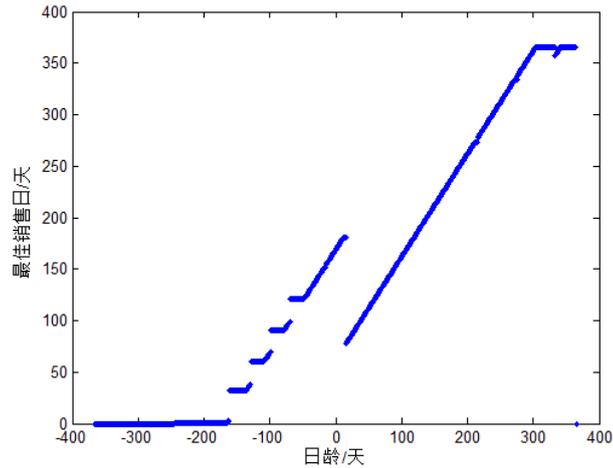


图 6.日龄-最佳销售日期

上图横坐标表示某只猪的日龄，纵坐标表示售卖该日龄的猪的最佳日期。分析上图可知，日龄小于-165 天的猪的最佳销售日期都是 1 月 1 日，因为这种类型的猪每天需要消耗饲料成本，但体重却不增长，因此越早卖出，收益越大。而日龄为-165 天到 15 天的猪的最佳售卖日期与日龄略成线性，大约都是在猪 160-170 天的时候卖掉，收益最大。而这一段也有售卖日期不随日龄变化的区间，这是由于每月副产品价格波动导致的。日龄 16 天到 303 天的猪的最佳销售日期与日龄也略成线性，大约是猪 60 天左右的时候卖掉，收益最大。造成这种现象的原因可以由图 7 解释：

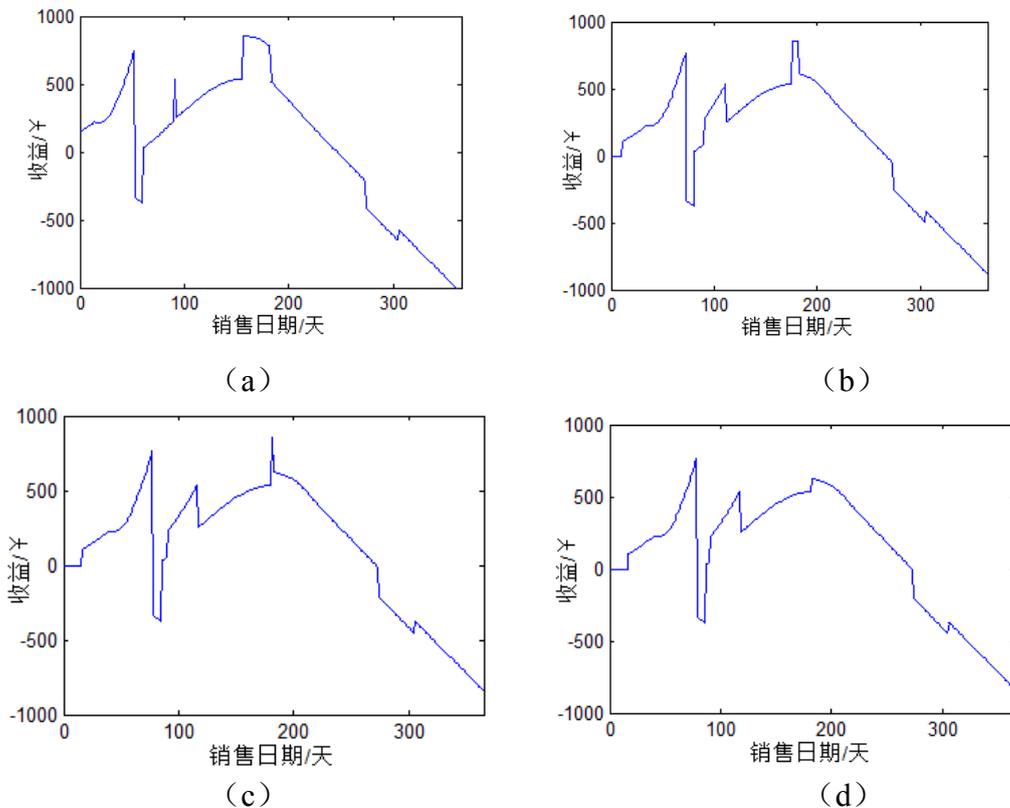


图 7 不同日龄的猪在全年售卖的收益 (a) 日龄 = -10 天 (b) 日龄 = -10 天 (c) 日龄 = 15 天 (d) 日龄 = 16 天

上图为不同日龄的猪在全年售卖的收益，可以看出，由于价格波动和猪体重增加而变为不同种类副产品的综合影响，售卖单只猪的收益在全年并非线性或是均匀的变化，而是呈现出多个局部峰值的变化。日龄为 15 的猪在第 181 天恰好由副产品三变为副产品四，所以收益在这一天有了激增，而 181 天为 6 月 30 日，182 天为 7 月 1 日，七月副产品四的价格较 6 月相比有一个较大的下降，因此收益在 182 天又有一锐减，这就造成了日龄为 15 的猪在 181 天的收益最大。而日龄为 16 的猪在 182 天综合了这两种变化，所以在该日期附近收益变化比较小。日龄大于 16 天的猪，由于错过了在 6 月以副产品四的高价卖出的机会，从而这些猪在体重小于 20kg 前，以副产品一的价格在日龄 60 天左右卖出的收益最高。

综合以上分析，结合图 6，在每一周售卖副产品猪时，优先选择最佳销售日期恰好在这一周的猪，便可在确定售卖总数的情况下，使得每一周的售卖收益最大。

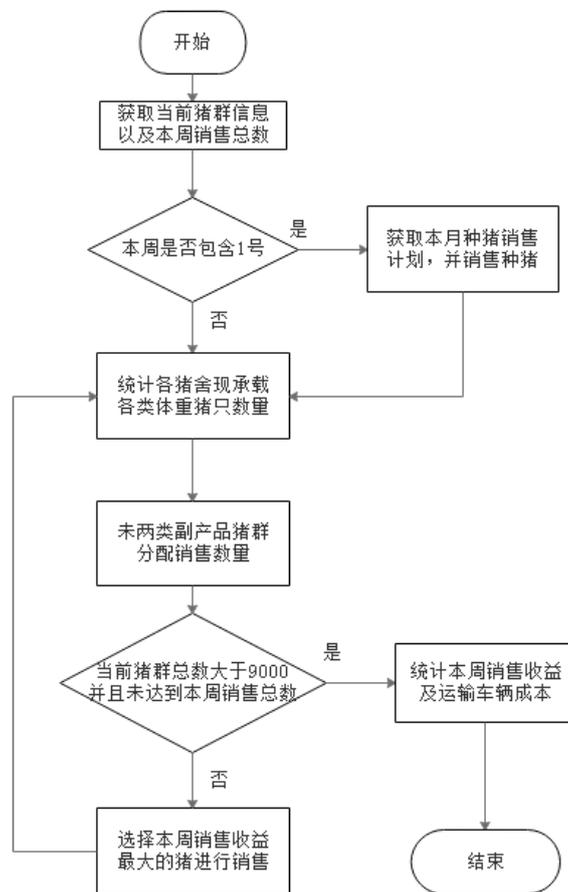


图 8. 销售计划制定流程图

为了验证本销售利益最大化策略确实能够增大总的销售收益，我们分别进行了两组实验，第一组实验在某一周售卖副产品猪时，采取随机销售策略，即随机选择符合条件的猪进行售卖。第二组实验优先选择最佳销售日期

恰好在这一周的猪，即采用本销售利益最大化策略。这两组实验所得到的年度累计销售收益如图 9 所示：

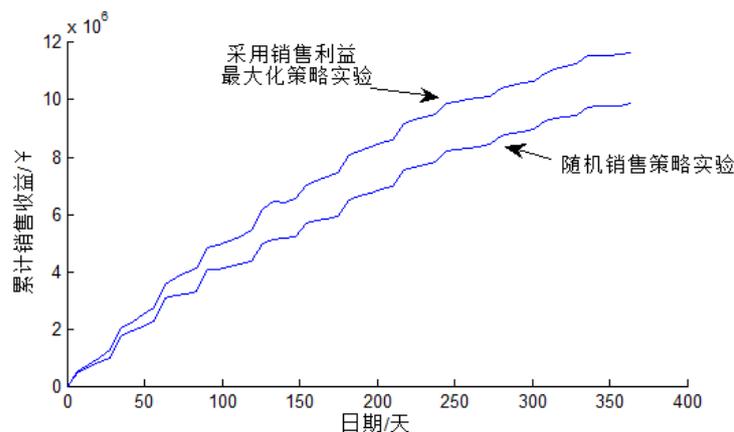


图 9. 不同销售策略全年累计销售收益对比图

第一组实验的全年累计销售收益为 988.78 万元，而第二组采用了销售利益最大化策略的实验的全年累计收益为 1162.23 万元，该结果明显优于随机销售策略实验，证明了本销售利益最大化策略的确优化了销售收益。

由第二组实验得到全年的猪群各种类猪的数量演化过程，如图 10 所示：

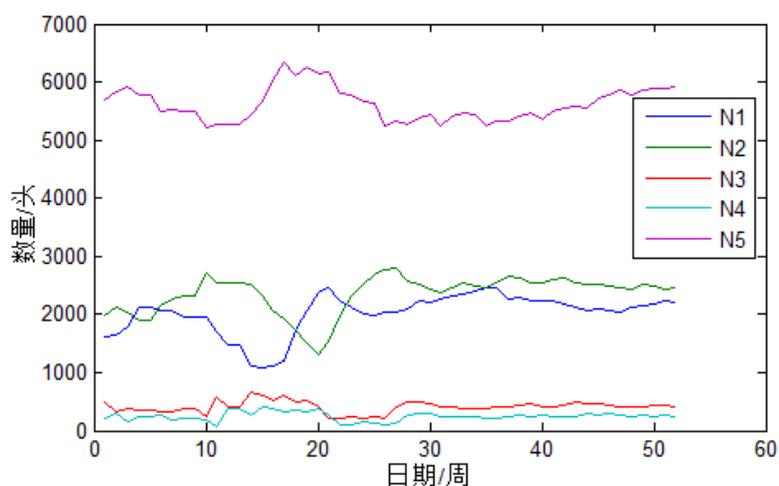


图 10. 猪群各种类猪的数量全年演化过程

由上图可以看出，各种类的猪的数量基本保持稳定，满足连续生产的要求。

5.3 粒子群算法优化

粒子群优化是计算智能领域的一种群体智能优化算法。其从生物种群行为特征中得到启发并利用与求解优化问题。算法中每个粒子都代表问题的一个潜在解。每个粒子对应一个由适应度函数决定的适应度函数值。粒子的速度决定了粒子的移动方向和距离。速度随着自身及其他粒子的移动经验进行动态调整。从而实现个体在可解空间的寻优。

(一) 问题描述

在本问题中，需要优化的参数是进行随机配种的条件下，各个种类的猪的配种概率 P_A P_B P_C P_D P_E P_F 。适应度函数是在不同的配种概率下，一年内的销售纯利润。函数如下：

$$f_{fitness} = profit - cost \quad (53)$$

在每次迭代过程中，粒子通过个体值和群体极值更新自己的速度和位置，如下式：

$$V_{ki}^{t+1} = \omega V_{ki}^t + c_1 r_1 (P_{kg,max}^t - P_{ki}^t) + c_2 r_2 (P_{ki,max}^t - P_{ki}^t) \quad (54)$$

$$P_{ki}^{t+1} = P_{ki}^t + V_{ki}^{t+1} \quad (55)$$

其中， V_{ki}^{t+1} 为粒子当前的速度， ω 为惯性权重。 $i=1,2,3,\dots$ 为粒子序号； $k=1,2,3,\dots$ 表示不同的参数类型，即配种概率类型。 c_1, c_2 为加速因子。 r_1, r_2 为 $[0,1]$ 的随机数。 $P_{ki,max}^t$ 表示个体极值， $P_{kg,max}^t$ 表示全局极值。

(二) 解题思路和步骤

流程图如下：

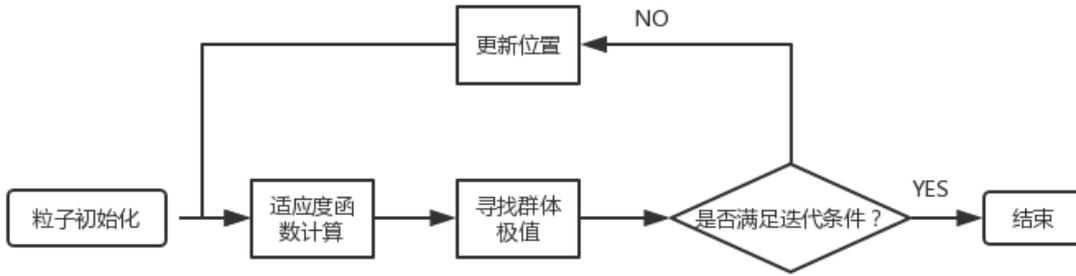


图 11. 粒子群算法流程图

其中粒子初始化即在满足约束的条件下，随机给粒子定初始值。根据公式(53)确定粒子的适应度函数值。依据公式(54)和公式(55)更新粒子位置。初始粒子在六维空间表示为

$$X = [P_A \ P_B \ P_C \ P_D \ P_E \ P_F] \quad (56)$$

其中 $\forall P, 0 < P < 1, P_A + P_B + P_C + P_D + P_E + P_F = 1$ 。在这样一个六维空间里，我们设计了 5 个粒子，迭代 20 次进行优化。

(三) 结果及分析

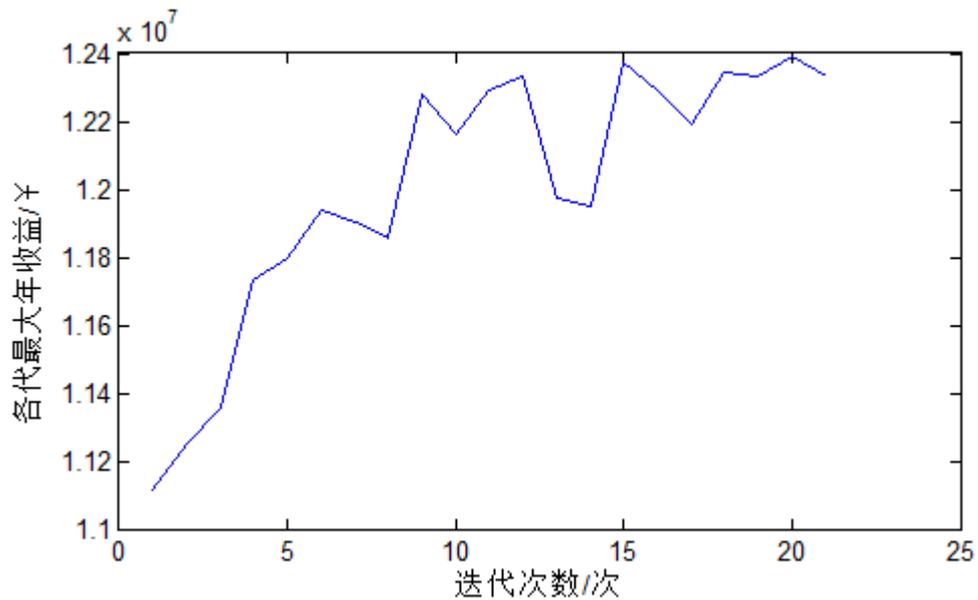


图 12. 粒子群算法优化迭代图

从上面的优化结果可以看出，其最大收益在 1240 万左右。在优化刚开始时，采用随机的方式选取粒子参数。随着迭代次数的增加，各代最大年收益的大体变化趋势是在逐步增加的。由于每代最优的粒子参数也会改变，因此趋势线中某些位置会有下降。当迭代次数到达 15 代以后，各代最优的收益值也趋于稳定，说明粒子逐渐收敛。从曲线的总体趋势上看，其粒子群算法的优化效果较为理想。

从上述的最优粒子中得到参数优化结果：

$$X = [0.0301 \quad 0.0987 \quad 0.1955 \quad 0.1028 \quad 0.1529 \quad 0.42]$$

将优化后的参数代入仿真流程，计算得到最终的配种计划和销售计划。该次仿真的年累计收益如下图所示：

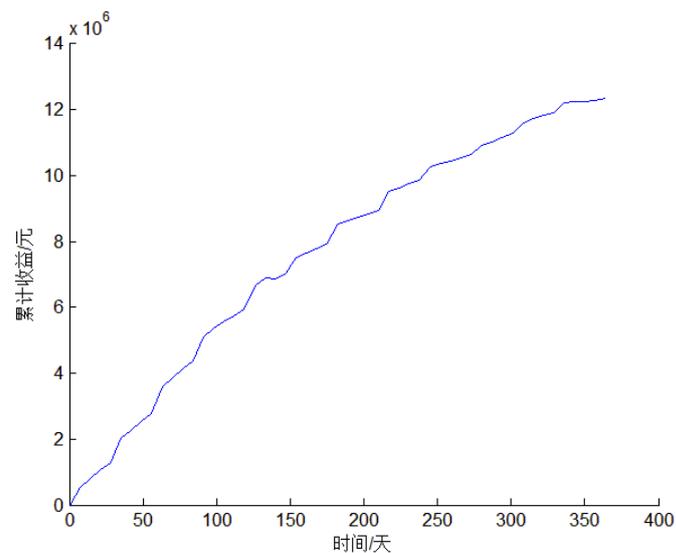


图 13. 年累积收益图

从图中可知该年收益为 1230.91 万，相比于未优化前数据（图 9 中所示）年收益明显提高。

5.4 饲料消耗量计算

在猪场仿真进行过程中，分别统计饲喂不同种饲料的猪的数量，再计算每种饲料的日消耗量，累加后即可得到一年内饲料的消耗情况，如下图所示：

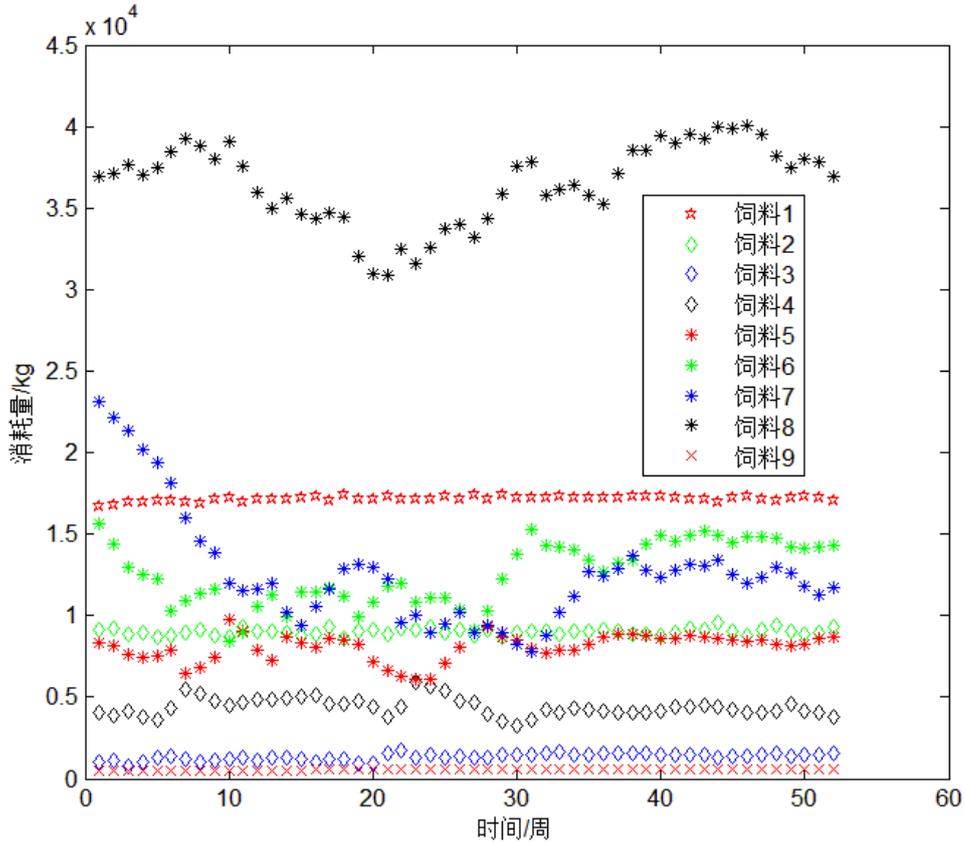


图 14. 各个饲料的消耗曲线图

由上图可以看出，各类饲料消耗量在一年内总体上没有太大的浮动，说明各个类型的猪的数量没有太大变化，种群结构基本保持稳定。图中最明显的变化是饲料 7 的消耗量在 0-15 周内迅速下降。这与销售计划有关。在前 15 周，对日龄为 101 到 130 天的猪的销售量较大，在 15 周以后，种群内每周的出生量和销售量达到了一个比较稳定的状态。

六、总结

6.1 模型的优点

(1) 在数据分析时，本文充分考虑了问题特点，依照现实情况对数据进行先聚类后拟合处理，为后期模型建立奠定基础。例如，根据附件 2 生长数据，我们发现在日龄为 120 天之后的重量数据较为分散，无法进行直接拟

合。实际情况为同样日龄下，育肥猪和种猪的体重具有较大差异。基于此，我们对数据集进行了聚类处理，并分别对其进行拟合，得到较为合理的不同种类猪日龄-重量拟合函数。

(1) 在模型建立过程中，本文创造性地以不同时期处于生长各个日龄的不同种类猪数量作为变量，建立了混合整数规划模型。在找到各个变量间的内在联系后，本文将原问题拆分为五个事件（母猪分娩、猪死亡、种猪出售、副产品出售、生产猪替换）、三个约束（场地约束，运输约束，存栏约束）和一个目标（最大化利益），使原问题得到简化。

(2) 在模型求解过程中，本文将一个多变量复杂问题进行拆分和简化，实现模型的分布求解。其总体求解流程为，先利用启发式算法制定配种策略和销售策略。利用粒子群优化算法对配种策略中的参数进行优化，并且采用销售收益最大化策略来优化销售计划，使得在两种计划的配合下，能够产生最大收益。

(3) 本文以天为步进单位，实现了配种计划与销售计划的精细化仿真与调度。最后设计实验对求解算法和优化策略进行验证：对销售收益最大化策略，实验表明，该策略能将年收益提高 14.92% 的同时，实现各种类猪的数量保持基本稳定；对粒子群算法，实验表明该算法能在 20 代粒子以内，将配种计划的收益提升 20.95%。

6.2 模型的缺点

(1) 模型在一些条件上进行了假设，对某些信息考虑不足，模型还不够完善。

(2) 模型有些条件都采取了理想化的处理，实际中应该更加细致的讨论

附录：代码（MATLAB）

%%粒子群优化

```
clear all;  
N = 10 ; % 种群规模  
D = 5 ; % 粒子维度  
T = 20 ; % 迭代次数  
a_xmax = 0.3 ;  
a_xmin = 0.01 ;  
b_xmax = 0.3 ;  
b_xmin = 0.01;  
c_xmax = 0.3 ;  
c_xmin = 0.01;  
d_xmax = 0.3 ;  
d_xmin = 0.01;  
e_xmax = 0.6 ;  
e_xmin = 0.01;  
v_max=0.1;  
v_min=-0.1;
```

```
C1 = 2 ; % 学习因子 1  
C2 = 2 ; % 学习因子 2  
W = 0.75 ; % 惯性权重
```

```
popx=zeros(N,D);  
popx(:,1) = rand(N,1)*(a_xmax-a_xmin)+a_xmin ; % 初始化粒子群的  
位置(粒子位置是一个 D 维向量)
```

```
popx(:,2) = rand(N,1)*(b_xmax-b_xmin)+b_xmin ;  
popx(:,3) = rand(N,1)*(c_xmax-c_xmin)+c_xmin ;  
popx(:,4) = rand(N,1)*(d_xmax-d_xmin)+d_xmin ;  
popx(:,5) = rand(N,1)*(e_xmax-a_xmin)+e_xmin ;
```

```
popx(1,:)=[0.4 0.1 0.1 0.1 0.1];
```

```
popv=zeros(N,D);  
popv(:,1) = rand(N,1)*(v_max-v_min)+v_min ; % 初始化粒子群的速度  
(粒子速度是一个 D 维度向量)
```

```
popv(:,2) = rand(N,1)*(v_max-v_min)+v_min ;  
popv(:,3) = rand(N,1)*(v_max-v_min)+v_min ;  
popv(:,4) = rand(N,1)*(v_max-v_min)+v_min ;  
popv(:,5) = rand(N,1)*(v_max-v_min)+v_min ;  
% 初始化每个历史最优粒子
```

```

pBest = popx ;
for i=1:N
pBestValue(i) = pig_farm(pBest(i,:)) ;
end

t_popx(:,1)=popx;
t_value(:,1)=pBestValue;
tBest(1) = max(pBestValue);
%初始化全局历史最优粒子
[gBestValue,index] = max(pBestValue) ;
gBest = popx(index,:) ;
for t=1:T
    fprintf('第%d代\n',t);
    for i=1:N
        fprintf('第%d个粒子\n',i);
        % 更新个体的位置和速度
        popv(i,:) =
W*popv(i,)+C1*rand*(pBest(i,)-popx(i,))+C2*rand*(gBest-popx(i,)) ;
        if sum(gBest==popx(i,))==5
            popv(i,)=zeros(1,D);
        end
        popx(i,) = popx(i,)+popv(i,) ;
        % 边界处理, 超过定义域范围就取该范围极值
        for jj=1:5
            if popv(i,jj)>v_max || popv(i,jj)<v_min
                popv(i,jj) = rand*(v_max-v_min)+v_min ;
            end
        end
    end

    if popx(i,1)>a_xmax || popx(i,1)<a_xmin
        popx(i,1) = rand*(a_xmax-a_xmin)+a_xmin ;
    end

    if popx(i,2)>b_xmax || popx(i,2)<b_xmin
        popx(i,2) = rand*(b_xmax-b_xmin)+b_xmin ;
    end

    if popx(i,3)>c_xmax || popx(i,3)<c_xmin
        popx(i,3) = rand*(c_xmax-c_xmin)+c_xmin ;
    end

    if popx(i,4)>d_xmax || popx(i,4)<d_xmin
        popx(i,4) = rand*(d_xmax-d_xmin)+d_xmin ;
    end

    if popx(i,5)>e_xmax || popx(i,5)<e_xmin
        popx(i,5) = rand*(e_xmax-b_xmin)+e_xmin ;
    end

```

```

end

% 计算目标函数，更新粒子历史最优
value(i)=pig_farm(popx(i,:));
if value(i)>pBestValue(i)
    pBest(i,:) = popx(i,:) ;
    pBestValue(i) = value(i);
end
if pBestValue(i) > gBestValue
    gBest = pBest(i,:) ;
    gBestValue = pBestValue(i) ;
end
fprintf('目标函数值: %5.5fn',value(i));
end
% 每代最优解对应的目标函数值
t_popx(:,t+1)=popx;
t_value(:,t+1)=value;
tBest(t+1) = max(value);
fprintf('每代最优: %d',max(value));
end

```

```
save('pso_workhouse');
```

```
%%猪场函数
```

```
function in_money=pig_farm(pop)
```

```
a=pop(1);
```

```
b=pop(2);
```

```
c=pop(3);
```

```
d=pop(4);
```

```
e=pop(5);
```

```
f=1-a-b-c-d-e;
```

```
if
```

```
a>0&&a<1&&b>0&&b<1&&c>0&&c<1&&d>0&&d<1&&e>0&&e<1&&f>0&&f<1
```

```
try
```

```
load init_data;
```

```
global time;
```

```
global hybrid_plan;
```

```
time=1;
```

```
my_sell=zeros(52,4);
```

```

%age_structure=zeros(52,6);
consum=zeros(364,9);
in_money=0;
prob=[a a+b a+b+c a+b+c+d a+b+c+d+e 1];
month=[31 59 90 120 151 181 212 243 273 304 334 365];
sale_plan2=sale_deal(sale_plan);
while(time<=364)
    %time

    %% 计算需要的销售种猪数量并配种
    [pig_all,sale_plan2,opp]=hybrid(pig_all,sale_plan2,month);

    %% 剩余配种机会随机配种
    if opp>0
        [pig_all]=rand_hybrid(pig_all,prob,opp);
    end

    %% 如果是非计划生产则按概率并插入总群    time<116
    temp=6;
    if time<=116
        for i=1:temp
            temp_pro=rand();
            for j=1:6
                if temp_pro<prob(j)
                    kind=j;
                    break;
                end
            end
            [pig_all,flag]=instant_born(pig_all,j);
            if flag==false
                temp=temp+1;
            end
        end
    end

    %% 出售并计算收益
    if mod(time ,7)==0

        pre_inmoney=in_money;
        [pig_all,num]=pre_sale(pig_all);

[pig_all,temp_money,temp_sell,temp_structure]=SELL_pig(pig_all,num,sale_pla
n);

```

```

        in_money=in_money+temp_money;
        my_sell(time/7,:)=temp_sell;
        %age_structure(time/7,:)=temp_structure;

        hold on;
        line([time-7,time],[pre_inmoney,in_money]);
        pause(0.00001);

    end
    %% 计算当天饲料消耗
%       [temp_consum]=feed(pig_all);
%       consum(time,:)=temp_consum;

    %% 记录
%       if time==364
%       save hybrid_plan hybrid_plan;
%       save my_sell my_sell;
%       save all;
%       end
    %% 时间步进
    time=time+1;

    %% 更新（设置死亡概率）(注意时间问题)

    pig_all=renew(pig_all,time);
end
catch
    in_money=0;
end

else
    in_money=0;
end

```

```

function [num_boar,num_sow]=sum_boarandsow(pig)

```

```

m=size(pig,1);
num_boar=0;
num_sow=0;

```

```

for i=1:m
    if pig(i,6)==1&&pig(i,5)==5
        num_boar=num_boar+1;
        continue;
    end
    if pig(i,6)==1&&pig(i,5)==4
        num_sow=num_sow+1;
        continue;
    end
end
end
end

```

```

function [Pig_new,Profit,sell_weight,a] = SELL_pig(Pig_now,S,sale_plan)

```

```

a=zeros(1,6);

```

```

l = length(Pig_now);

```

```

for i = 1:l

```

```

    Pig_now(i,1) = i;

```

```

end

```

```

Profit = 0;

```

```

aready_sell = 0;

```

```

sell_weight = [0,0,0,0];

```

```

global time;

```

```

S = S -100;

```

```

fir_day = [1,32,60,91,121,152,182,213,244,274,305,335];

```

```

ls = ismember(fir_day,time-6);

```

```

if day2month(time-6)~=day2month(time)|| sum(ls)~=0    %如果是 1 号，要先把
这个月的种猪处理掉

```

```

    month = day2month(time);

```

```

    if sale_plan(1,month)~=0    %卖 A 系公猪

```

```

        sell_num = 0;

```

```

        i = 1;

```

```

        while sell_num < sale_plan(1,month);

```

```

            if Pig_now(i,2)==1 && Pig_now(i,4)==1 && Pig_now(i,5)==3 &&

```

```

Pig_now(i,6)==1 %如果可以卖

```

```

                Pig_now(i,:) = [];

```

```

                sell_num = sell_num + 1;

```

```

                aready_sell = aready_sell + 1;

```

```

            else

```

```

                i = i + 1;

```

```

            end

```

```

        end

```

```

    end

```

```

    if sale_plan(3,month)~=0    %卖 C 系公猪
        sell_num = 0;
        i = 1;
        while sell_num < sale_plan(3,month);
            if Pig_now(i,2)==3 && Pig_now(i,4)==1 && Pig_now(i,5)==3 &&
Pig_now(i,6)==1 %如果可以卖
                Pig_now(i,:) = [];
                sell_num = sell_num + 1;
                aready_sell = aready_sell + 1;
            else
                i = i + 1;
            end
        end
    end
end
if sale_plan(5,month)~=0    %卖 E 系公猪
    sell_num = 0;
    i = 1;
    while sell_num < sale_plan(5,month);
        if Pig_now(i,2)==5 && Pig_now(i,4)==1 && Pig_now(i,5)==3 &&
Pig_now(i,6)==1 %如果可以卖
            Pig_now(i,:) = [];
            sell_num = sell_num + 1;
            aready_sell = aready_sell + 1;
        else
            i = i + 1;
        end
    end
end
end
if sale_plan(2,month)~=0    %卖 B 系母猪
    sell_num = 0;
    i = 1;
    while sell_num < sale_plan(2,month);
        if Pig_now(i,2)==2 && Pig_now(i,4)==0 && Pig_now(i,5)==3 &&
Pig_now(i,6)==1 %如果可以卖
            Pig_now(i,:) = [];
            sell_num = sell_num + 1;
            aready_sell = aready_sell + 1;
        else
            i = i + 1;
        end
    end
end
end
if sale_plan(4,month)~=0    %卖 D 系母猪
    sell_num = 0;

```

```

    i = 1;
    while sell_num < sale_plan(4,month);
        if Pig_now(i,2)==4 && Pig_now(i,4)==0 && Pig_now(i,5)==3 &&
Pig_now(i,6)==1 %如果可以卖
            Pig_now(i,:) = [];
            sell_num = sell_num + 1;
            aready_sell = aready_sell + 1;
        else
            i = i + 1;
            if i == length(Pig_now)
                break;
            end
        end
    end
end
end
if sale_plan(6,month)~=0 %卖 F 系母猪
    sell_num = 0;
    i = 1;
    while sell_num < sale_plan(6,month);
        if Pig_now(i,2)==6 && Pig_now(i,4)==0 && Pig_now(i,5)==3 &&
Pig_now(i,6)==1 %如果可以卖
            Pig_now(i,:) = [];
            sell_num = sell_num + 1;
            aready_sell = aready_sell + 1;
        else
            i = i + 1;
        end
    end
end
end
end

if S > aready_sell
    S = S - aready_sell;
else
    Pig_new = Pig_now;
    return;
end

NA = 3377;
NB = 7588;
A=[];
B=[];
BR = 0;
BY_x = 0;

```

```
BY_y = 0;
YF_x = 0;
YF_y = 0;
```

```
for i=1:length(Pig_now)
    age = Pig_now(i,3);
    if age<28
        BR = BR + 1;
    elseif age>27 && age<67
        BY_x = BY_x + 1;
        if Pig_now(i,6) == 2
            A = [A,Pig_now(i,1)];
        end
    elseif age>66 && age<77
        BY_y = BY_y + 1;
    elseif age>76 && age<83
        YF_x = YF_x + 1;
    else
        YF_y = YF_y + 1;
        if Pig_now(i,6) == 2
            B = [B,Pig_now(i,1)];
        end
    end
end
end
```

```
if BY_x<NA &&YF_y<NB
    inA = (NA-BY_x)/NA;
    inB = (NB-YF_y)/NB;
    SA = round(inB*S/(inA+inB));
    SB = S-SA;
end
```

```
for i=1:SA
    while 1
        rd = 1+round(rand*(length(A)-1));
        sell_ID = A(rd);
        row = find(Pig_now(:,1)==sell_ID);
        if (Pig_now(row,7)<20 || Pig_now(row,7)>25)
            A(rd) = [];
            break;
        end
    end
end
```

```

if Pig_now(row,7)<=20
    sell_weight(1) = sell_weight(1) + Pig_now(row,7);
else
    sell_weight(2) = sell_weight(2) + Pig_now(row,7);
end
Profit = Profit + js_profit(Pig_now(row,3));
Pig_now(row,:)=[];
end

```

```

for i=1:SB
    if numel(B)==0
        hhh=0;
    end
    rd = 1+round(rand*(length(B)-1));

    sell_ID = B(rd);
    B(rd) = [];
    row = find(Pig_now(:,1)==sell_ID);
    if (Pig_now(row,7)>=25 && Pig_now(row,7) <50)
        sell_weight(2) = sell_weight(2) + Pig_now(row,7);
    elseif Pig_now(row,7)>=50 && Pig_now(row,7) <100
        sell_weight(3) = sell_weight(3) + Pig_now(row,7);
    else
        sell_weight(4) = sell_weight(4) + Pig_now(row,7);
    end
    Profit = Profit + js_profit(Pig_now(row,3));
    Pig_now(row,:)=[];
end

```

```

n = length(Pig_now);
vector = randperm(n);
Pig_new = zeros(size(Pig_now));
for i = 1:n
    Pig_new(i,:) = Pig_now(vector(i,:));
    Pig_new(i,1)=i;
end

```

```

a=[BR,BY_x,BY_y,YF_x,YF_y,n];

```

```

end

```

```

function plan_new=sale_deal(plan_old)

```

```

%将销售计划数量转变为窝数

```

```

for i=1:6
    for j=1:12
        if i<=4
            plan_new(i,j)=plan_old(i,j)/0.98/0.97/0.90/0.88/5;
        end
        if i==5
            plan_new(i,j)=plan_old(i,j)/0.98/0.97/0.90/0.88/0.4/9;
        end
        if i==6
            plan_new(i,j)=plan_old(i,j)/0.98/0.97/0.90/0.88/0.65/11;
        end
    end
end
end
end

```

```

function pig_new=renew(pig_old,time)
%时间往后更新一天，更新猪的状态
%pig_old 中每列分别表示 ID\ 种类\ 日龄\ 公 1 母 2 \状态（分类如下）\ 类别
（1 种猪 2 副产品）\ 重量\ 生产 flag（0 1）
% 更新的类别主要是 日龄 状态 重量
% 哺乳仔猪 1、保育猪 2、待售种猪 3、生产母猪 4、生产公猪 5、后备母猪 6
[m,n]=size(pig_old);
pig_order=0;
[num_boar,num_sow]=sum_boarandsow(pig_old);
for i=1:m
    pig_order=pig_order+1;
    %% 判断死亡
    pro=1;
    switch pig_old(pig_order,3)
        case 11
            pro=0.9;
        case 46
            pro=0.97;
        case 119
            pro=0.98;
    end
    if pro~=1
        if rand()>pro
            pig_old=die(pig_old,pig_order);
            pig_order=pig_order-1;
            continue;
        end
    end
end
end

```

```

end

%% 更新日龄
pig_old(pig_order,3)=pig_old(pig_order,3)+1;

%% 更新状态
if pig_old(pig_order,3)==28
    pig_old(pig_order,5)=2;
end
if pig_old(pig_order,3)==77
    pig_old(pig_order,5)=3;
end
%% 更新类别
% E 选种失败的成为副产品
if
pig_old(pig_order,2)==5&&pig_old(pig_order,4)==1&&pig_old(pig_order,3)==1
61
    if rand()>0.60
        pig_old(pig_order,6)=2;
    end
end
% F 选种失败的成为副产品
if
pig_old(pig_order,2)==6&&pig_old(pig_order,4)==0&&pig_old(pig_order,3)==1
05
    if rand()>0.35
        pig_old(pig_order,6)=2;
    end
end
% ABCD 选种失败的成为副产品
if
pig_old(pig_order,2)~=6&&pig_old(pig_order,2)~=5&&pig_old(pig_order,4)==0
&&pig_old(pig_order,3)==147
    if rand()>0.35
        pig_old(pig_order,6)=2;
    end
end
%生产公猪的替换
if
pig_old(pig_order,2)<=4&&pig_old(pig_order,6)==1&&pig_old(pig_order,4)==1
&&pig_old(pig_order,5)==5&&pig_old(pig_order,3)>666
    pig_old(pig_order,6)=2;
    for j=1:size(pig_old,1)

```



```

        pig_old(pig_order,6)=2;
    end
end

    if num_boar-35<0
        if
pig_old(pig_order,4)==1&&pig_old(pig_order,6)==1&&pig_old(pig_order,2)~=6
&&pig_old(pig_order,2)~=5&&pig_old(pig_order,3)>=301
            pig_old(pig_order,5)=5;
            num_boar=num_boar+1;
        end
    else
        if
pig_old(pig_order,5)~=5&&pig_old(pig_order,4)==1&&pig_old(pig_order,6)==1
&&pig_old(pig_order,2)~=6&&pig_old(pig_order,2)~=5&&pig_old(pig_order,3)
>=501
            pig_old(pig_order,6)=2;
        end
    end

    %% 更新重量
    t=pig_old(pig_order,3);

    if t<24
        pig_old(pig_order,7)=5+0.089*t;
    end
    if t>=24&&t<192
        pig_old(pig_order,7)=-0.00004163*t^3+0.01368*t^2-0.6092*t+14.7;
    end
    if t>=192
        pig_old(pig_order,7)=107.208+0.001*t;
    end
    %% 生小猪
    if pig_old(pig_order,8)==time
        pig_old=born(pig_old,pig_old(pig_order,9));
    end
end

```

end

pig_new=pig_old;

end

function pig_new=rand_hybrid(pig_old,prob,opp)

global time;

global hybrid_plan;

while opp>0

temp=rand();

for i=1:6

if temp<prob(i)

kind=i;

break;

end

end

[pig_old,flag]=choose_mo(pig_old,kind);

if flag==true

hybrid_plan(time,opp)=kind;

opp=opp-1;

end

%pig_old=born(pig_old,kind);

end

pig_new=pig_old;

end

function [pig_new, num] = pre_sale(pig_old)

plan=[16 200 200; 3 600 480; 8 200 54 ;3 300 150;300 0 0 ;500 0 0];

m=size(pig_old,1);

for i=1:m

if pig_old(i,6)==2

continue;

end

% E 剔除多余

if pig_old(i,2)==5&&pig_old(i,3)>=147

plan(5,1)=plan(5,1)-1;

if plan(5,1)<0

pig_old(i,6)=2;

end

```

end
% F 剔除多余
if pig_old(i,2)==6&&pig_old(i,3)>=105
    plan(6,1)=plan(6,1)-1;
    if plan(6,1)<0
        pig_old(i,6)=2;
    end
end
end
% AC 类待售剔除
if (pig_old(i,2)==1||pig_old(i,2)==3)
    if pig_old(i,4)==1&&pig_old(i,5)==3&&pig_old(i,3)>=77
        plan(pig_old(i,2),3)=plan(pig_old(i,2),3)-1;
        if plan(pig_old(i,2),3)<0
            pig_old(i,6)=2;
        end
    end
end
end
% BD 类待售剔除
if pig_old(i,2)==2||pig_old(i,2)==4
    if pig_old(i,4)==0&&pig_old(i,5)==3&&pig_old(i,3)>=77
        plan(pig_old(i,2),3)=plan(pig_old(i,2),3)-1;
        if plan(pig_old(i,2),3)<0
            pig_old(i,6)=2;
        end
    end
end
end
end

end
num=0;
for i=1:m
    if pig_old(i,3)<=7
        num=num+1;
    end
end

pig_new=pig_old;
end

```

```

function [profit] = js_profit(age)
time = 3;
load Cost_price_weight;
month = day2month(time);

```

```

cost = Cost(age);
weight = Weight(age);

if weight<=20
    price = Price(4,month)*weight;
elseif weight>20 &&weight<=25
    price = 0;
elseif weight>25 &&weight<=50
    price = Price(3,month)*weight;
elseif weight>50 &&weight<=100
    price = Price(2,month)*weight;
else
    price = Price(1,month)*weight;
end

```

```

profit = price - cost;

```

```

end

```

```

function [pig_new, flag]=instant_born(pig_old,kind)

```

```

global time;

```

```

flag=false;

```

```

m=size(pig_old,1);

```

```

if kind==5

```

```

    mo_kind=4;

```

```

end

```

```

if kind==6

```

```

    mo_kind=2;

```

```

end

```

```

if kind<5

```

```

    mo_kind=kind;

```

```

end

```

```

for i=1:m

```

```

    if

```

```

        pig_old(i,6)==1&&pig_old(i,2)==mo_kind&&pig_old(i,3)>=210&&pig_old(i,5)=
        =4&&pig_old(i,8)==0

```

```

            pig_old(i,8)=time;

```

```

            pig_old(i,9)=kind;

```

```

            pig_old=born(pig_old,kind);

```

```

            flag=true;

```

```

            break;

```

```

        end

```

```
end
pig_new=pig_old;
end
```

```
function [pig_new,plan_new,opp]=hybrid(pig_old,plan_old,month)
global time;
global hybrid_plan;
opp=6;
```

```
% 选择月份
month_flag=false;
for i=1:12
    if time+277<=month(i)
        mon=i;
        month_flag=true;
        break;
    end
end
```

```
while month_flag==true&&opp>0&&plan_old(5,mon)>0
    [pig_old,flag]=choose_mo(pig_old,5);
    if flag==false
        break;
    end
    %pig_old=born(pig_old,5);
    plan_old(5,mon)=plan_old(5,mon)-1;
    hybrid_plan(time,opp)=5;
    opp=opp-1;
end
```

```
month_flag=false;
for i=1:12
    if time+221<=month(i)
        mon=i;
        month_flag=true;
        break;
    end
end
```

```
while month_flag==true&&opp>0&&plan_old(6,mon)>0
    [pig_old,flag]=choose_mo(pig_old,6);
    if flag==false
        break;
    end
end
```

```

        end
        %pig_old=born(pig_old,6);
        plan_old(6,mon)=plan_old(6,mon)-1;
        hybrid_plan(time,opp)=6;
        opp=opp-1;
    end

    month_flag=false;
    for i=1:12
        if time+194<=month(i)
            mon=i;
            month_flag=true;
            break;
        end
    end
    for i=1:4
        while month_flag==true&&opp>0&&plan_old(i,mon)>0
            [pig_old,flag]=choose_mo(pig_old,i);
            if flag==false
                break;
            end
            %pig_old=born(pig_old,i);
            plan_old(i,mon)=plan_old(i,mon)-1;
            hybrid_plan(time,opp)=i;
            opp=opp-1;
        end
    end

    plan_new=plan_old;
    pig_new=pig_old;

```

end

```

function [temp_consum_kg]=feed(pig_all)
temp_consum=zeros(1,9);
temp_consum_kg=zeros(1,9);
m=size(pig_all,1);
con_kg=[2.6 5.12 0.3 0.6 1.1 1.6 2.25 2.4 2.5];
for i=1:m
    if pig_all(i,5)==4&&pig_all(i,6)==1
        if rand()<0.79

```

```

        temp_consum(1,1)=temp_consum(1,1)+1;
    else
        temp_consum(1,2)=temp_consum(1,2)+1;
    end
    continue;
end
if pig_all(i,5)==5&&pig_all(i,6)==1
    temp_consum(1,9)=temp_consum(1,9)+1;
    continue;
end
if pig_all(i,3)>=21&&pig_all(i,3)<=28
    temp_consum(1,3)=temp_consum(1,3)+1;
    continue;
end
if pig_all(i,3)>=29&&pig_all(i,3)<=45
    temp_consum(1,4)=temp_consum(1,4)+1;
    continue;
end
if pig_all(i,3)>=46&&pig_all(i,3)<=70
    temp_consum(1,5)=temp_consum(1,5)+1;
    continue;
end
if pig_all(i,3)>=71&&pig_all(i,3)<=100
    temp_consum(1,6)=temp_consum(1,6)+1;
    continue;
end
if pig_all(i,3)>=101&&pig_all(i,3)<=130
    temp_consum(1,7)=temp_consum(1,7)+1;
    continue;
end
if pig_all(i,3)>=131
    temp_consum(1,8)=temp_consum(1,8)+1;
    continue;
end
end
end

for i=1:9
    temp_consum_kg(1,i)=temp_consum(1,i)*con_kg(i);
end

```

```

function pig_new=die(pig_old,order)

```

```

m=size(pig_old,1);

```

```

% if order>m
%     fprintf('die wrong');
%     return;
% end

pig_old(order,:)=[];

for i=order:m-1
    pig_old(i,1)=pig_old(i,1)-1;
end

pig_new=pig_old;
end

function [month] = day2month(day)
if day <32
    month = 1;
elseif 31<day && day<60
    month = 2;
elseif 59<day && day<91
    month = 3;
elseif 90<day && day<121
    month = 4;
elseif 120<day && day<152
    month = 5;
elseif 151<day && day<182
    month = 6;
elseif 181<day && day<213
    month = 7;
elseif 212<day && day<244
    month = 8;
elseif 243<day && day<274
    month = 9;
elseif 273<day && day<305
    month = 10;
elseif 304<day && day<335
    month = 11;
else
    month = 12;
end
end

```

```
function [pig_new,flag]=choose_mo(pig_old,kind)
```

```
global time;
```

```
flag=false;
```

```
m=size(pig_old,1);
```

```
if kind==5
```

```
    mo_kind=4;
```

```
end
```

```
if kind==6
```

```
    mo_kind=2;
```

```
end
```

```
if kind<5
```

```
    mo_kind=kind;
```

```
end
```

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

```
    if
```

```
    pig_old(i,6)==1&&pig_old(i,2)==mo_kind&&pig_old(i,3)>=210&&pig_old(i,5)=  
    =4&&(pig_old(i,8)<time-24||pig_old(i,8)==0)
```

```
        pig_old(i,8)=time+116;
```

```
        pig_old(i,9)=kind;
```

```
        %pig_old=born(pig_old,kind);
```

```
        flag=true;
```

```
        break;
```

```
    end
```

```
end
```

```
pig_new=pig_old;
```

```
end
```

```
function [ k_cf,k_by, k_yf ] = check_house( pig )
```

```
% 输入猪的信息，输出空的产床数量，保育室栏位数量，和育肥室栏位数量
```

```
% 此处显示详细说明
```

```
    cf=144;
```

```
    by=168;
```

```
    yf=336;
```

```
    cf_pig=0;
```

```
    by_pig1=0;
```

```
    by_pig2=0;
```

```
    yf_pig1=0;
```

```
    yf_pig2=0;
```

```
    m=size(pig,1);
```

```

for i=1:m
    if pig(i,3)<=24
        continue;
    end

    if pig(i,8)==1
        cf_pig=cf_pig+1;
        continue;
    end

    if pig(i,7)<=22.7
        by_pig1=by_pig1+1;
        continue;
    end
    if pig(i,7)<25
        by_pig2=by_pig2+1;
        continue;
    end

    if pig(i,7)<=34
        yf_pig1=yf_pig1+1;
        continue;
    else
        yf_pig2=yf_pig2+1;
        continue;
    end
end

k_cf=cf-cf_pig;
k_by=by-by_pig1/24-by_pig2/18;
k_yf=yf-yf_pig1/46-yf_pig2/23;

```

end

```

function pig_new=born(pig_old,kind)
    m=size(pig_old,1);

    if kind <=4
        num=10;
    end

    if kind ==5
        num=19;
    end

```

```

end

if kind ==6
    num=23;
end
kid=0;
for i=1:num
    if rand()>0.88
        continue;
    end
    kid=kid+1;
    pig_old(m+kid,1)=m+kid;
    pig_old(m+kid,2)=kind;
    pig_old(m+kid,3)=1;
    if rand()>0.5
        pig_old(m+kid,4)=0;
    else
        pig_old(m+kid,4)=1;
    end
    pig_old(m+kid,5)=1;
    % 类别确定
    pig_old(m+kid,6)=1;

    if kind==5&&pig_old(m+kid,4)==0
        pig_old(m+kid,6)=2;
    end
    if kind==6&&pig_old(m+kid,4)==1
        pig_old(m+kid,6)=2;
    end
    if (kind==2||kind==4)&&pig_old(m+kid,4)==1
        if rand()<0.9
            pig_old(m+kid,6)=2;
        end
    end
end

    pig_old(m+kid,7)=5.089;
    pig_old(m+kid,8)=0;
    pig_old(m+kid,9)=0;
    pig_old(m+kid,10)=0;

end
pig_new=pig_old;

end

```