五一数学建模竞赛

承　诺　书

我们仔细阅读了五一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）与本队以外的任何人（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它公开的资料（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，我们愿意承担由此引起的一切后果。

我们授权五一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）。

参赛题号（从 A/B/C 中选择一项填写）：A

参赛队号：T20734636998656

参赛组别（研究生、本科、专科、高中）：本科

所属学校（学校全称）：四川大学锦城学院

参赛队员：队员 1 姓名：饶志龙

队员 2 姓名：魏鑫瑶

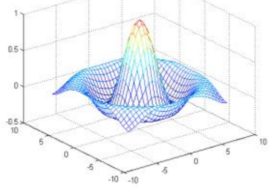
队员 3 姓名：胥程

联系方式：Email：rzl1379343390@163.com 联系电话：19102859813

日期：2021 年 5 月 3 日

（除本页外不允许出现学校及个人信息）

五 一 数 学 建 模 竞 赛



题　目：　　　　疫苗生产问题

关键词：动态规划模型 Palmer 算法 蒙特卡洛法 甘特图

摘　要：

本文通过动态规划模型、Palmer 算法、蒙特卡洛法、绘制甘特图等方法算出了疫苗斜度指标、生产时间均值、方差、最值、概率分布、安排顺序等，建立了疫苗生产模型，并分析了如何安排疫苗生产才能最快完成任务或获得最大销售额。

针对问题一，要求通过每箱疫苗在所有工位上的生产时间进行均值、方差、最值、概率分布等统计分析出每个工位生产疫苗的能力水平。本文将所给数据导入 MATLAB 软件中，使用 str2num 函数进行数据类型转换，再分别用 mean 函数、std 函数、max 函数、min 函数算出均值、方差、最值，并用 plot 函数绘制平均时间图像、用 distribution fitter 对数据进行概率分布拟合，以便分析。

针对问题二，要求建立数学模型，为 YM1-YM10 各 100 剂疫苗的加工制定生产顺序，计算生产总时间。本文建立动态规划模型，使用 Palmer 算法算出疫苗斜度指标并得到所求顺序：YM4、YM2、YM1、YM7、YM8、YM10、YM5、YM3、YM6、YM9，并用 MATLAB 得到每种疫苗进入 CJ1 时刻的时间与离开 CJ4 时刻的时间与所需总时间 210.9165 分钟。

针对问题三，要求建立数学模型，在问题二所得时间基础上减少 5%，以最大概率完成任务为目标来确定生产顺序，并给出缩短时间比例与最大概率的关系。本文利用蒙特卡洛法进行疫苗生产总时间的随机模拟，统计比问题 2 所需总时间减少 5%的样本数量，计算出其占比，即为最大概率。再通过更改总时间减少比例，重复操作，可得所求关系。

针对问题四，要求每个工位每天生产时间不超过 16 小时，每种类型疫苗生产任务不能拆分，建立数学模型，使完成任务的可靠性为 90%。本文利用所给数据求出完成所有疫苗加工的平均时间并结合问题一中所求方差求出答案范围，再运用蒙特卡洛法在 MATLAB 软件进行 1000 组数据模拟，取第 901 条数据并结合范围分析得出至少需要 598.9774 天可以完成任务。

针对问题五，要求以最大销售额为目标，建立数学模型，并在每个工位每天生产时间不超过 16 小时的情况下进行 100 天内的部分疫苗生产。本文求得完成每种疫苗任务所需时间与销售额，继而得出单位时间内生产每种疫苗所得销售额，选择单位时间内可得销售额高的疫苗进行生产则可在 100 天内得到最高利润，即安排生产 6 万剂 YM9、9 万剂 YM10 和 20.8507 万剂 YM4 可得最高利润 1011.29341 万美元。

一、问题重述

疫苗生产需按固定顺序经过 4 个不同工艺流程进行加工，每个工艺流程一次性均能处理 100 剂疫苗且每个工位不能同时生产不同类型的疫苗，也不能插队。现需生产 10 种不同类型的疫苗，模拟后发现，因生产设备、疫苗纯化等原因，各工位生产不同类型的每箱疫苗用时不稳定。为探讨工位生产疫苗的能力水平，制定完成疫苗生产任务的方案，我们建立模型解决以下问题：

问题一：根据所给数据，对每箱疫苗在所有工位上的生产时间进行均值、方差、最值、概率分布等统计分析，掌握每个工位生产疫苗的能力水平。

问题二：已知每个工位生产每箱疫苗平均时间，以在最短时间内交付 YM1-YM10 疫苗各 100 剂为目标，建立数学模型，制定疫苗生产顺序，计算生产总时间，并将结果填入表中。

问题三：若要求该公司交货时间比问题二的总时间缩短 5%，建立数学模型，以最大的概率完成此任务为目标来确定生产顺序，并给出缩短时间比例与最大概率之间的关系。

问题四：已知 10 种疫苗不同规模的生产任务，每个工位每天生产时间不超过 16 小时，每种类型疫苗生产任务不能拆分，建立数学模型，安排生产方案，使完成任务的可靠性为 90%。

问题五：已知每种疫苗出厂价格、每个工位生产每箱疫苗平均时间，每个工位每天生产时间不超过 16 小时，每种类型疫苗的生产任务可拆分，以最大销售额为目标，建立数学模型安排生产计划。

二、模型假设

(1)假设在疫苗生产中不会出现意外导致疫苗损耗。
(2)假设问题四中，每个工位每天生产时间都为 16 小时。
(3)假设问题四、五中不考虑由于工序时间间隔导致的疫苗完整性。
(4)假设问题五中不考虑刚开始 YM9 在 CJ1 时，YM10 的等待时间。

三、符号说明

序号	符号	意义
1	λ_i	疫苗的斜度指标
2	i	疫苗序号
3	m	工位序号
4	p_{im}	疫苗 i 在工位CJ_m上的加工时间
5	f_t	制定的生产顺序中排在第一位的疫苗的加工时间
6	wt_{jk}	制定的生产顺序中排在第 j 位的疫苗在第 k 个工位的等待时间
7	j	制定的生产顺序的序号

四、问题分析

4.1 问题一的分析

问题一要求根据每箱疫苗在每个工位上的生产平均时间、方差、最值、以及概率分布进行分析，因此我们利用 MATLAB 软件，将数据进行导入并通过相应函数得解，再对图表进行分析得出不同工位生产疫苗的能力水平。

4.2 问题二的分析

问题二要求根据每个工位生产每箱疫苗平均时间制定疫苗生产顺序，达到时间最短的目的。从问题一中我们已求得每个工位生产每箱疫苗的平均时间，则可建立动态规划模型，运用 Palmer 算法算出疫苗的斜度指标，再按照疫苗的斜度指标进行降序排列，从而得到所求顺序并用 MATLAB 得到所求时间。

4.3 问题三的分析

问题三要求该公司疫苗交货时间在问题二所得时间的基础上缩短 5%，并以最大概率完成任务为目标来确定生产顺序，给出缩短时间比例与最大概率的关系。因为生产时间具有随机性，所以采用蒙特卡洛法进行随机模拟。再统计比问题 2 所得结果总时间缩短 5%的样本数量，并计算占比，该占比即为该情况下的最大概率。再更改缩短的时间比例并进行同样的操作，即可得到所求关系。

4.4 问题四的分析

问题四要求根据已安排的生产疫苗任务来安排生产方案，建立数学模型，约束每个工位每天生产时间不超过 16 小时，每种类型疫苗生产任务不能拆分，并使完成任务的可靠性为 90%。可以通过生产所有疫苗的平均时间和方差确定结果的大致范围，再采用蒙特卡洛法进行随机模拟，得出所求结果。

4.5 问题五的分析

问题五要求以最大销售额为目标，建立数学模型，在 100 天内选择部分疫苗进行生产，每个工位每天生产的时间不超过 16 小时，每类疫苗生产任务可拆分。由问题四可得完成每种疫苗任务所需时间，利用所给数据可求得完成每种任务可得销售额。将每种疫苗的销售额除以所需时间得单位时间内生产每种疫苗所得的销售额，并将此进行降序排列，选择排位高的部分疫苗进行生产即可得到最高利润。

五、模型建立与求解

5.1 问题一求解

5.1.1 求每箱疫苗在各个工位上的生产时间均值

将所给表格中的数据导入 MATLAB 中，先通过 str2num 函数将字符数据转换为数值数据，再使用 mean 函数得出所求均值，再使用 plot 函数绘制图形进行观察分析，得出结论。如表 1 所示：

表 1 每箱疫苗在各个工位上的生产时间均值

疫苗种类 \ 工位	CJ1	CJ2	CJ3	CJ4
YM1	13.2840	14.9621	19.8460	20.0129
YM2	9.8709	19.9075	17.9282	18.9424
YM3	20.0584	15.9726	14.9704	15.1164
YM4	7.9887	9.9366	5.9359	18.1284
YM5	8.7701	13.7220	13.0052	11.2495
YM6	19.0741	20.0944	14.1485	13.8839

YM7	11.1601	16.4961	12.0137	19.0876
YM8	16.0201	8.8275	18.1144	16.8314
YM9	15.0146	12.0351	7.0419	8.9497
YM10	12.9524	7.0110	9.0492	16.0524

CJ1、CJ2、CJ3、CJ4 制作 YM1~YM10 的平均时间分别如图 1、图 2、图 3、图 4 所示：

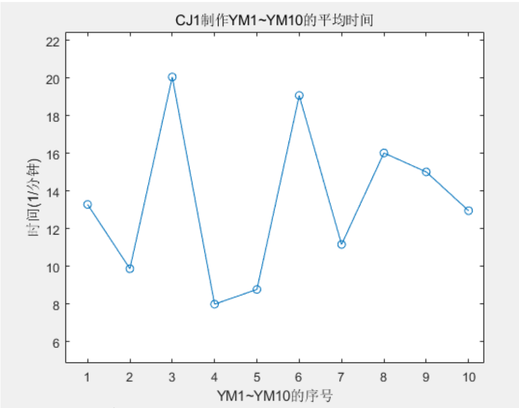


图 1 CJ1 制作 YM1~YM10 的平均时间

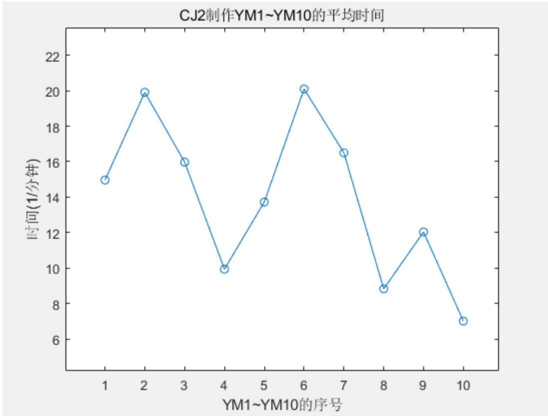


图 2 CJ2 制作 YM1~YM10 的平均时间

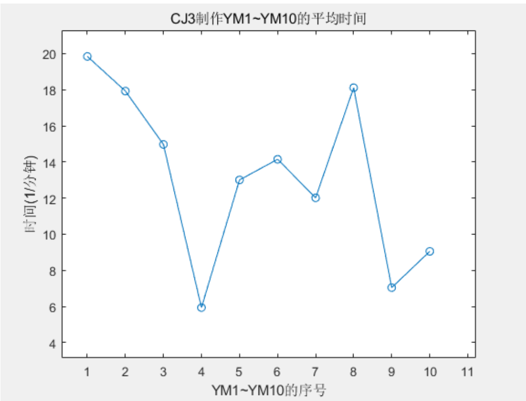


图 3 CJ3 制作 YM1~YM10 的平均时间

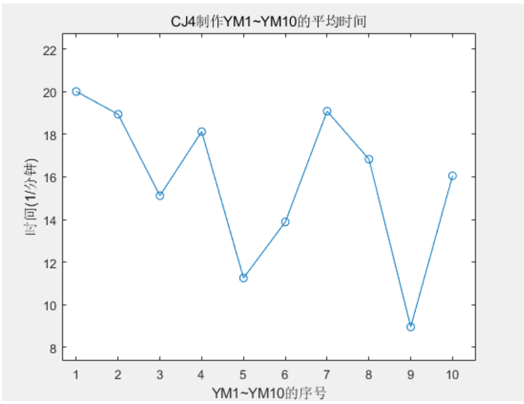


图 4 CJ4 制作 YM1~YM10 的平均时间

5.1.2 求每箱疫苗在各个工位上的生产时间方差

使用 std 函数求出转换后的数值数据的均方差，将均方差进行平方得到方差。如表 2 所示：

表 2 每箱疫苗在各个工位上的生产时间方差

工位 / 疫苗种类	CJ1	CJ2	CJ3	CJ4
YM1	1.59397926684082	1.08308077290204	1.18750462297551	1.88973229562857
YM2	0.833284616326531	1.25291127780000	0.902185019628572	0.890484535546938
YM3	0.844790781636735	0.693394179285715	1.04003727642449	0.812172135363265
YM4	0.932171112751021	0.953390967840816	0.0379964340571429	1.12771557856735
YM5	0.74508907342	1.13957072352	0.83130109054	1.23082773071

	4490	653	6939	429
YM6	1.33498554367347	0.973471332689796	0.875459281404082	1.15173114542449
YM7	1.02630680091429	0.929460647024489	0.786215724800000	0.656362277126531
YM8	1.13172521102041	0.261407526530612	1.07001134177959	0.912768772412245
YM9	0.973807391220409	1.15807468857143	0.133662711840816	0.175195030289796
YM10	0.207491463363265	0.282183874159184	0.207364087857143	0.263394333636735

5.1.3 求每箱疫苗在各个工位上的生产时间最值

使用 max 函数求出每箱疫苗在各个工位上的生产时间的最大值，如表 3 所示：

表格 3 每箱疫苗在各个工位上的生产最大值

工位 疫苗种类	CJ1	CJ2	CJ3	CJ4
YM1	16.5784	17.5855	22.908	23.789
YM2	11.6821	22.2294	20.1778	21.4245
YM3	21.814	17.8089	17.7873	17.0034
YM4	10.0389	12.5088	6.3889	20.7891
YM5	10.3845	16.6052	15.0108	14.5267
YM6	21.7526	21.9278	15.8136	16.9491
YM7	13.2878	18.7304	14.4366	20.7553
YM8	18.0783	10.1871	21.5699	18.7351
YM9	17.4124	14.7485	7.9947	9.8984
YM10	13.6186	8.5793	9.8172	17.4872

使用 min 函数求出每箱疫苗在各个工位上的生产时间最小值，如表 4 所示：

表格 4 每箱疫苗在各个工位上的生产时间最小值

工位 疫苗种类	CJ1	CJ2	CJ3	CJ4
YM1	10.0557	13.067	17.8616	16.5052
YM2	6.9708	17.5137	15.5031	17.1641
YM3	17.814	13.978	12.671	12.8065
YM4	5.872	7.6405	5.3536	16.0432
YM5	5.927	11.7967	10.7985	9.2316
YM6	16.4356	17.5005	12.0908	11.7249
YM7	8.8435	14.6801	9.6528	16.7402
YM8	13.5753	7.9676	16.3152	14.7471
YM9	12.8237	10.0932	6.2319	8.1086
YM10	11.4639	5.9503	7.9742	14.9504

5.1.4 求每箱疫苗在所有工位上的生产时间概率分布

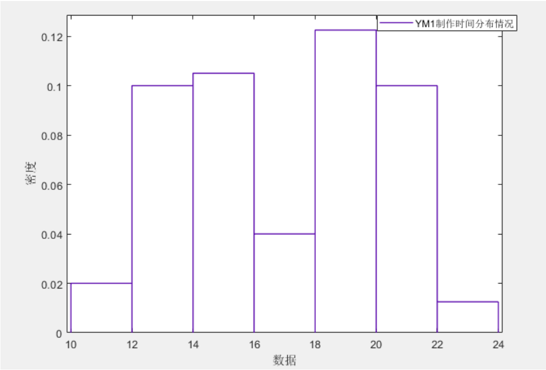


图 5YM1 在所有工位上的生产时间概率分布

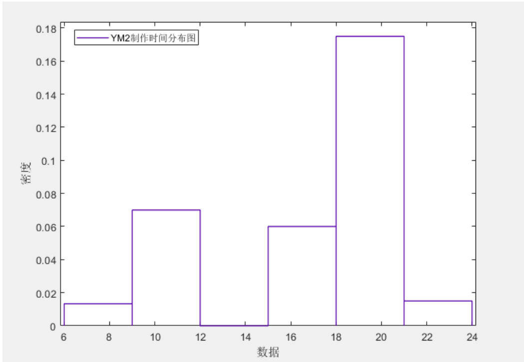


图 6YM2 在所有工位上的生产时间概率分布

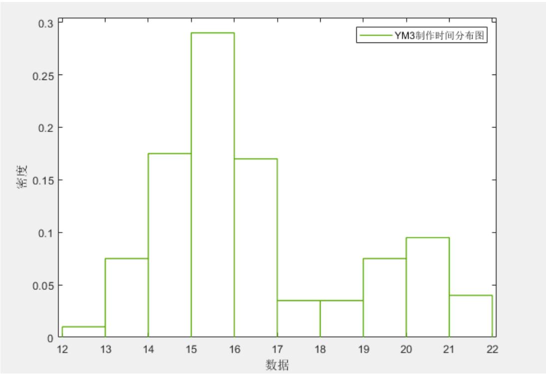


图 7YM3 在所有工位上的生产时间概率分布

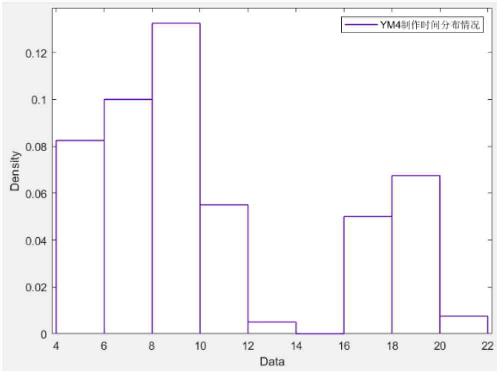


图 8YM4 在所有工位上的生产时间概率分布

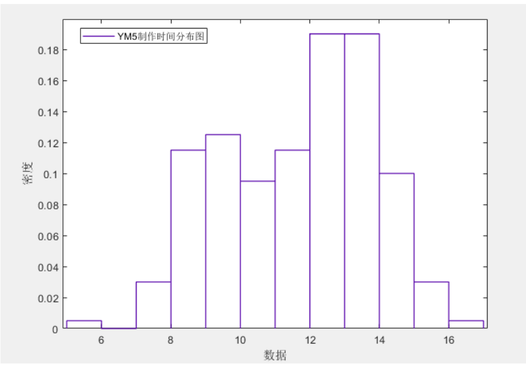


图 9YM5 在所有工位上的生产时间概率分布

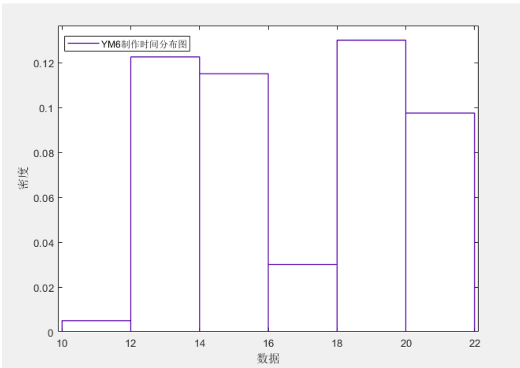


图 10YM6 在所有工位上的生产时间概率分布

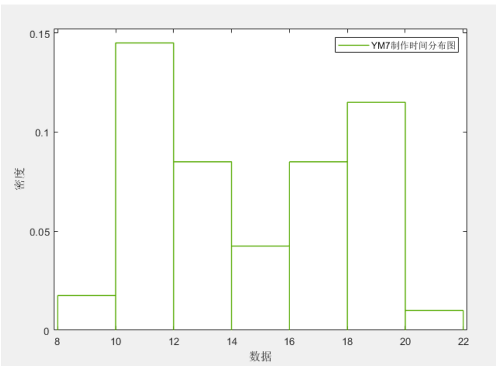


图 11YM7 在所有工位上的生产时间概率分布

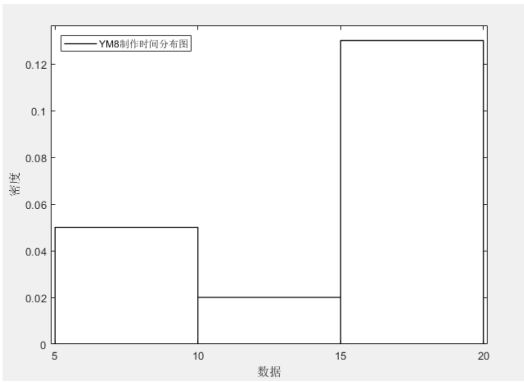
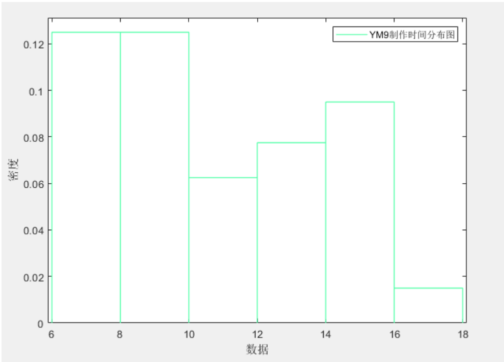


图 12YM8 在所有工位上的生产时间概率分布



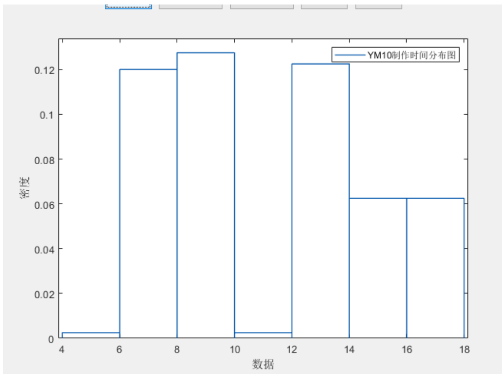


图 13YM9 在所有工位上的生产时间概率分布 图 14YM10 在所有工位上的生产时间概率分布

根据上方所有分析可供疫苗生产参考。

5.2 问题二模型建立与求解

5.2.1 模型的建立

通过以上分析，我们知道：疫苗生产过程中存在第 j 个疫苗在第 k 个工位未加工完成而导致的第(j+1)个疫苗在第 k-1 个工位等待的情况。因此采用动态规划模型，建立目标函数$f(x)_{min}$=f_t+$\sum_{j=2}^{10} wt_{jk}$，$f(x)_{min}$为所求最短疫苗生产总时间，f_t为排序后最先生产的疫苗生产所需时间，wt_{jk}表示第 j 个生产的疫苗在第 k 个工位等待所需时间，k=1,2,3,4。两者相加即为所求疫苗生产总时间。

5.2.2 模型的求解

运用 Palmer 算法[1]，根据斜度指标计算公式$\lambda_i = \sum_{j=1}^{m}[j-(m+1)/2]\times p_{ij}$[2]求出疫苗 YM1~YM10 的斜度指标，如表 5 所示：

表格 5 YM1-YM10 的斜度指标

疫苗序号	YM1	YM2	YM3	YM4	YM5
斜度指标λ	12.5353	12.61766	-7.9141	13.2092	3.3607
疫苗序号	YM6	YM7	YM8	YM9	YM10
斜度指标λ	-10.7583	9.6501	5.8604	-11.5940	5.6691

将 YM1-YM10 按照斜度指标进行降序排列，得出疫苗生产顺序为：YM4、YM2、YM1、YM7、YM8、YM10、YM5、YM3、YM6、YM9

通过 MATLAB 计算得出生产的总时间为：$f(x)_{min}$=210.9165（分钟）

利用甘特图进行时间的细化分析，如图 15 所示：

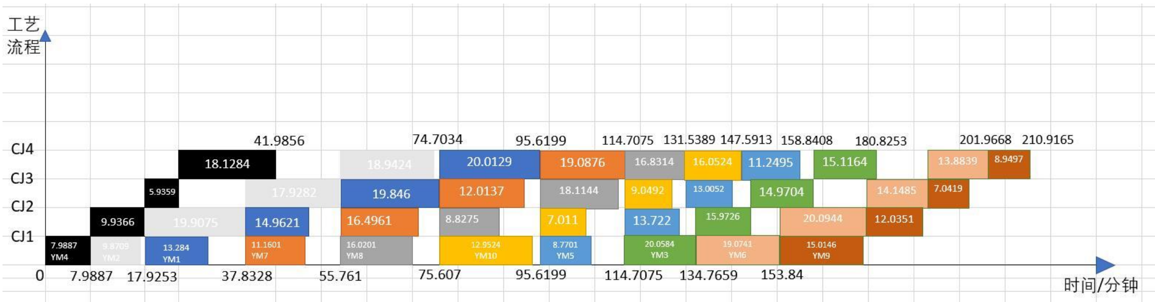


图 15 不同疫苗生产时间甘特图

疫苗的生产顺序以及每种类型疫苗进入 CJ1 的时间和离开 CJ4 的时间如表 6：

表 6 每种类型疫苗进入 CJ1 的时间和离开 CJ4 的时间

加工顺序（填疫苗编号）	进入 CJ1 时刻	离开 CJ4 时刻
YM4	00:00	41.9896
YM2	7.9887	74.7034
YM1	17.9223	95.6199
YM7	37.8328	114.7075
YM8	55.761	131.5389
YM10	75.607	147.5913
YM5	95.6199	158.8408
YM3	114.7075	180.8253
YM6	134.7659	201.9668
YM9	153.84	210.9165

5.3 问题三求解

5.3.1 模型的建立

由上述分析可得，每个工位生产每种疫苗的所需时间具有随机性，因此我们运用蒙特卡洛法[3]在 MATLAB 上进行 1000 组数据的随机模拟。选出比问题 2 的总时间缩短 5%的样本并进行数量统计，计算出的占比即为最大概率。

再不断对总时间减少的比例进行更改，并统计符合要求的样本占比，可得总时间缩短比例与最大概率之间的关系。

5.4 问题四求解

通过以上分析，先对问题一中已求得的每个工位生产每种疫苗所需平均时间进行加运算，得到不同疫苗加工完毕所需时间和天数如表 7：

表 7 不同疫苗加工所需时间

疫苗类型	箱	加工一箱所需时间（分钟）	所需总时间（小时）	所需天数
YM1	1000	68.105	1135.083333	70.94271
YM2	500	66.649	555.4083333	34.71302
YM3	600	66.1178	661.178	41.32363
YM4	1000	41.9896	699.8266667	43.73917
YM5	1200	46.7468	934.936	58.4335

YM6	1600	67.2009	1792.024	112.0015
YM7	1800	58.7575	1762.725	110.1703
YM8	800	59.7934	797.2453333	49.82783
YM9	600	43.0413	430.413	26.90081
YM10	900	45.065	675.975	42.24844
合计 Total	10000	563.4663	9444.814667	590.3009

利用上表算出的平均时间 590.3009 天和问题一求出的方差确认结果的大致范围。再运用蒙特卡洛法进行对每个工位生产每种疫苗所需时间的随机模拟，将模拟出的1000组数据进行升序排列，取第901组条数据与上述表格中所需天数进行分析，作为答案，即 598.9774。

5.5 问题五求解

5.5.1 模型的建立

通过以上分析，我们需要在问题四求得的天数基础上，再通过所给数据求出生产各种疫苗可得的总销售额。将生产每种疫苗可得到的总销售额除以生产每种疫苗需要的天数得一比例，选比例高者进行生产。即在单位时间内可得销售额低的疫苗将不会被生产。

5.5.2 模型的求解

利用已知数据算出生产各疫苗所需时间、可得的总销售额、比例（万美元/天），并将所有疫苗按照单位时间内可得销售额按照降序排列，如表 8 所示：

表格 8 生产每种疫苗所需时间、可得销售额与比例

序号	疫苗类型	任务数量（万剂）	出厂价格（美元/剂）	箱	万美元	所需天数（天）	万美元/天
1	YM9	6	46	600	276	26.9008125	10.25991
2	YM10	9	48	900	432	42.2484375	10.22523
3	YM4	10	43	1000	430	43.73916667	9.831006
4	YM5	12	42	1200	504	58.4335	8.625189
5	YM8	8	51	800	408	49.82783333	8.188195
6	YM7	18	48	1800	864	110.1703125	7.842403
7	YM2	5	54	500	270	34.71302083	7.778061
8	YM3	6	50	600	300	41.323625	7.25977
9	YM6	16	45	1600	720	112.0015	6.428485
10	YM1	10	42	1000	420	70.94270833	5.92027

所以得出：生产 YM9、YM10 与部分 YM4。

能获得的总销售额即为 276+432+(100-26.9008125-42.2484375)*9.83100577285804=1011.29341。即结果为：安排生产 6 万剂 YM9、9 万剂 YM10 和 30.8507 万剂 YM4，可得销售额为 1011.29341 万美元。

六、模型评价

1. 采用了动态规划模型和 Palmer 算法，制定出的疫苗生产顺序符合以最短时间交付的目标。
2. 使用了蒙特卡洛法随机模拟得出数据，使得出的结果更加科学、客观。

七、模型改进

1. 因资料搜集不够完整，数据还不够精确，理想化假设较多，实际化的可能不强。

八、模型的推广和应用

1. 该模型可以运用于工厂流水线生产，也可以应用于公司多层面试及其他有先后顺序的多级服务程序。

九、参考文献

[1] 龙蛋蛋君，多机作业排序问题一约翰逊算法和帕尔默法求最优解，https://www.cnblogs.com/litao1105/p/5140420.html，2021 年 5 月 2 日

[2] 糖果屋 123053，生产第十一章_制造业作业计划与控制，https://wenku.baidu.com/link?url=16I4Vg6F-KcVAT0DimalNMali3zViRx7Us8686jnSk5qVrfUqh2llG09o8YI6gGxg-7lzk4F6-B9fqU3Ybr_RHeERxYVMzBkJevzuBwHSvi，2021 年 5 月 2 日

[3] Acdreamers，蒙特卡洛算法，https://blog.csdn.net/acdreamers/article/details/44978591?%3E，2021 年 5 月 2 日

附录：

附录一：问题一中的 MATLAB 代码：

```
for i=1:40
    z1=str2num(z(i))
    zd(i,1)=max(z1(:))
    zx(i,1)=min(z1(:))
    zp(i,1)=mean(z1(:))
    zf(i,1)=var(z1(:))
end
```

附录二：问题二中的 MATLAB 代码：

```
function [Makespan,Schedule]=Palmer(PT)
[n,m]=size(PT);
if n<=1
    error('The job qty must large than 2')
end
```

```
for i=1:n
    SlopeIndex(i)=0;
    for j=1:m
        SlopeIndex(i)=SlopeIndex(i)+(2*j-m-1)*PT(i,j);
    end
end
[Best,BestIndex]=sort(SlopeIndex,'descend');
StartTime(1:m,1:n)=0;
StartTime(1,BestIndex(1))=0;
for j=2:n
StartTime(1,BestIndex(j))=StartTime(1,BestIndex(j-1))+PT(BestIndex(j-1),
1);
end
for k=2:m
StartTime(k,BestIndex(1))=StartTime(k-1,BestIndex(1))+PT(BestIndex(1),k-
1);
    for j=2:n
StartTime(k,BestIndex(j))=max(StartTime(k,BestIndex(j-1))+PT(BestIndex(j
-1),k),...
       StartTime(k-1,BestIndex(j))+PT(BestIndex(j),k-1));
    end
end
Makespan=StartTime(m,BestIndex(n))+PT(BestIndex(n),m);
Schedule=BestIndex;
```

附录三：问题三 MATLAB 代码：

```
x=[13.2840,1.59397926684082;14.9621,1.08308077290204;19.8460,1.187504622
97551;20.0129,1.88973229562857;9.8709,0.833284616326531;19.9075,1.252911
27780000;17.9282,0.902185019628572;

18.9424,0.890484535546938;20.0584,0.844790781636735;15.9726,0.6933941792
85715;14.9704,1.04003727642449;15.1164,0.812172135363265;7.9887,0.932171
112751021;9.9366,0.953390967840816;

5.9359,0.0379964340571429;18.1284,1.12771557856735;8.7701,0.745089073424
490;13.7220,1.13957072352653;13.0052,0.831301090546939;11.2495,1.2308277
3071429;19.0741,1.33498554367347;

20.0944,0.973471332689796;14.1485,0.875459281404082;13.8839,1.1517311454
2449;11.1601,1.02630680091429;16.4961,0.929460647024489;12.0137,0.786215
724800000;19.0876,0.656362277126531;

16.0201,1.1317252110;8.8275,0.2614075265;18.1144,1.07001134177;16.8314,0
.9127687724;15.0146,0.973807391220409;12.0351,1.15807468857143;7.0419,0.
133662711840816;8.9497,0.175195030289796;
```

```
12.9524,0.207491463363265;7.0110,0.282183874159184;9.0492,0.207364087857143;16.0524,0.263394333636735]
   for i=1:40
   y=x(i,1)+sqrt(x(i,2))*rand(5,10)
   w(i)=sum(y(:))/50
   end
 w(:,1);
```

附录四：问题四 MATLAB 代码：

```
pj=[13.2840,1.59397926684082;14.9621,1.08308077290204;19.8460,1.18750462297551;20.0129,1.88973229562857;9.8709,0.833284616326531;19.9075,1.25291127780000;17.9282,0.902185019628572;

18.9424,0.890484535546938;20.0584,0.844790781636735;15.9726,0.693394179285715;14.9704,1.04003727642449;15.1164,0.812172135363265;7.9887,0.932171112751021;9.9366,0.953390967840816;

5.9359,0.0379964340571429;18.1284,1.12771557856735;8.7701,0.745089073424490;13.7220,1.13957072352653;13.0052,0.831301090546939;11.2495,1.23082773071429;19.0741,1.33498554367347;

20.0944,0.973471332689796;14.1485,0.875459281404082;13.8839,1.15173114542449;11.1601,1.02630680091429;16.4961,0.929460647024489;12.0137,0.786215724800000;19.0876,0.656362277126531;

16.0201,1.1317252110;8.8275,0.2614075265;18.1144,1.07001134177;16.8314,0.9127687724;15.0146,0.973807391220409;12.0351,1.15807468857143;7.0419,0.133662711840816;8.9497,0.175195030289796;

12.9524,0.207491463363265;7.0110,0.282183874159184;9.0492,0.207364087857143;16.0524,0.263394333636735];
    total=0
    for r=1:1000
    for i=1:40
        if rand(1)<0.5
   z=pj(i,1)-sqrt(pj(i,2));
   sj(i)=z;
        else
   z=pj(i,1)+sqrt(pj(i,2));
   sj(i)=z;
        end
    end
   sj';
for i=1:10
```

```
    x(i)=sj((i-1)*4+1)+sj((i-1)*4+2)+sj((i-1)*4+3)+sj((i-1)*4+4);
end
y=[1000,500,600,1000,1200,1600,1800,800,600,900];
for i=1:10
    w(i)=(x(i)*y(i))/960;
end
 sum(w)
    end
```