数学实验：第五次作业

计算机系　计 73　2017011620　李家昊

2020 年 4 月 14 日

1　实验目的

- 掌握 MATLAB 优化工具箱的基本用法，对不同算法进行初步分析、比较。
- 练习用无约束优化方法建立和求解实际问题的模型（包括非线性最小二乘拟合）。
- 掌握用 MATLAB 优化工具箱和 LINGO 解线性规划的方法。
- 练习建立实际问题的线性规划模型。

2　问题求解

2.1　Chap7-Ex5 原子位置（应用题）

2.1.1　问题分析

题目给定某个分子的原子个数，以及某些原子对之间的距离，需要确定每个原子的相对位置。对于该问题，可以求出一组最优的原子相对位置，使得原子之间的距离尽可能接近给定的距离，即两者之间的误差尽可能小，这就构成了一个无约束优化问题。

2.1.2　模型假设

为了简化实际问题，模型基于以下假设，

1. 该分子为平面分子，所有原子处于同一平面上。
2. 该分子的结构稳定，所有原子之间的距离固定不变。
3. 给定的原子对距离是原子位置的唯一约束。

表 1: 五种方法求解得出的最优值，所需迭代次数及目标函数调用次数

Method	$z_{\min}$	Iterations	Func Count
SteepDesc	5.7759	548	53802
BFGS	0.0246	196	10143
DFP	0.0471	10001	491225
LM	0.0575	60	3023
TRM	0.0602	161	7938

2.1.3 模型建立

设原子个数为 n，不失一般性，不妨以第 n 个原子作为参考系，即第 n 个原子的坐标为 $(0,0)$，设第 i 个原子的坐标为 (x_i,y_i)（$i=1,2,\cdots,n-1$），题目给定的原子对距离约束 (i,j,d_{ij}) 构成集合 $\mathcal{D}$，其中 d_{ij} 表示第 i 个原子与第 j 个原子之间的距离，为了满足给定约束，原子距离需要尽可能满足下式，

$$\sqrt{(x_i-x_j)^2+(y_i-y_j)^2}=d_{ij},\quad (i,j,d_{ij})\in\mathcal{D} \tag{1}$$

在本题中，方程 (1) 确定了 52 个方程，但只包含 48 个变量，属于超定方程组，一般来说没有精确解，因此只能求出最小二乘意义下的最优解，其优化目标为，

$$\min z=\sum_{(i,j,d_{ij})\in\mathcal{D}}\left|\sqrt{(x_i-x_j)^2+(y_i-y_j)^2}-d_{ij}\right|^2 \tag{2}$$

方程 (2) 即为本题的模型。

2.1.4 算法设计

方程 (2) 是一个无约束优化问题，可采用最速下降法和拟牛顿法求解，对应的命令为 `fminunc`，同时它也是一个非线性最小二乘问题，也可采用 LM 方法和置信域方法求解，对应的命令为 `lsqnonlin`。

2.1.5 Matlab 程序

请参见附录4.1。

2.1.6 计算结果

将 48 个变量的初值置为全零，分别通过最速下降法（SteepDesc），拟牛顿法的 BFGS 公式，拟牛顿法的 DFP 公式，LM 方法，以及置信域（TRM）方法计算原子的位置，求解结果如表 1所示，表中列出了五种方法求出的最优值，所

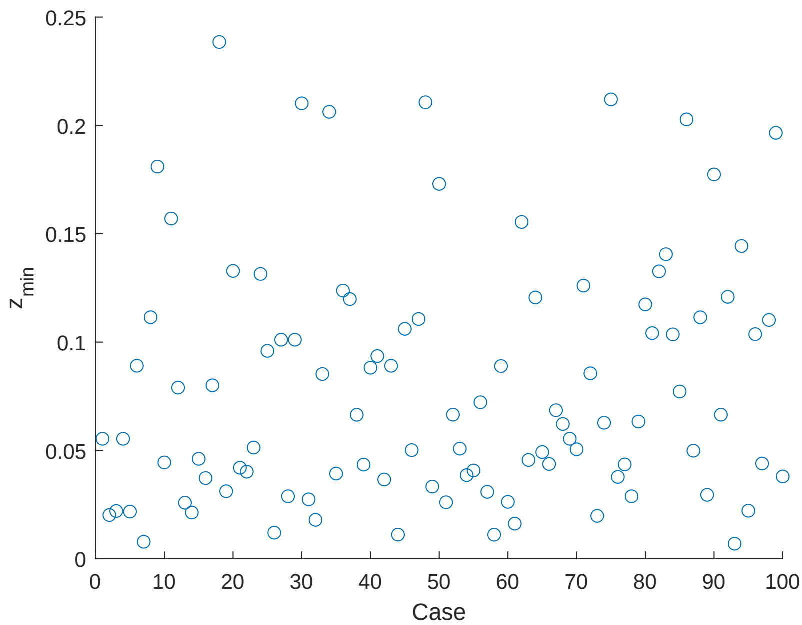

图 1: 多次随机选取初始值，采用 BFGS 公式求解得到的最优值

用的迭代次数，以及目标函数的调用次数，其中 DFP 公式因迭代次数超过最大值而提前终止。

为了进一步探索模型的稳定性，这里在区间 $(-1,1)$ 内取 48 个独立均匀分布的随机数，作为变量的初值，用 BFGS 公式进行求解，结果如图 1所示。

在上述的初值条件探索过程中，取出最优值最小的一组解，作为原子相对位置的最终结果，如图 2所示，其最优值为 0.0070，原子的相对坐标如表 2所示，其中只列出了前 24 个原子的坐标，第 25 个原子的坐标为 $(0,0)$。

表 2: 相对于第 25 个原子，第 1-24 个原子位置的数值结果

No.	x	y	No.	x	y	No.	x	y
1	0.7735	0.0720	9	0.0964	-0.3875	17	0.1195	-1.1353
2	0.1549	-0.7119	10	0.0648	0.3029	18	0.9269	0.0457
3	0.8619	0.3747	11	-0.4456	0.0281	19	-0.3240	-0.1804
4	-0.1729	0.2237	12	1.1031	-0.1822	20	0.1525	0.0113
5	0.0889	-0.1802	13	0.3306	-0.6111	21	-0.0429	1.0177
6	0.0164	0.3961	14	-0.2973	0.1729	22	-0.1914	0.7867
7	-0.0681	0.2586	15	0.0377	-0.1707	23	0.5830	0.6564
8	0.0855	-0.0659	16	-0.4258	-0.8661	24	-0.1383	0.9274

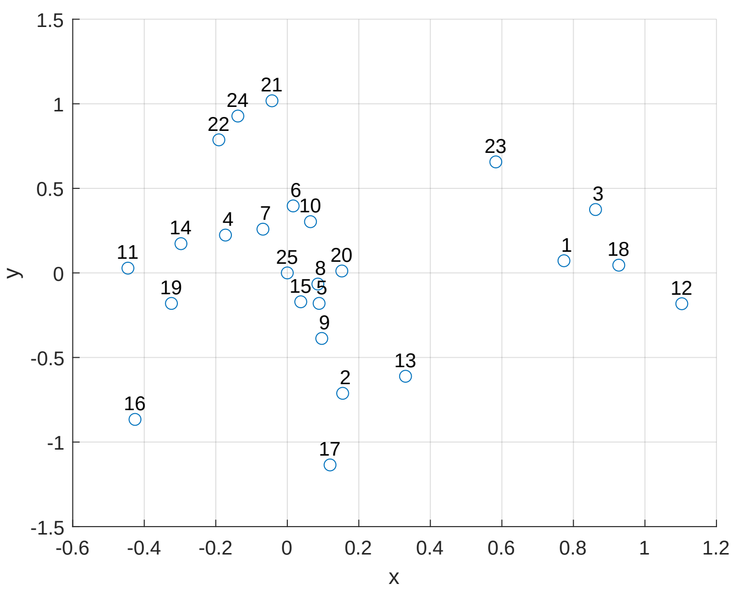

图 2: 相对于第 25 个原子，第 1-24 个原子的位置

2.1.7 结果的数学分析

不同方法的对比 由表 1可以看出，在全零初值条件下，最速下降法的结果最差，在本题条件下并不收敛，拟牛顿法的 BFGS 公式的精度最高，迭代次数较少，DFP 公式精度较高，但收敛较慢，迭代次数较多，LM 算法的迭代次数最少，精确度较高，TRM 方法相比 LM 算法精度较低，收敛较慢。在效率和精度的综合考虑下，对于该非线性无约束优化问题，BFGS 的求解性能最佳，因此这里采用 BFGS 公式来计算原子相对位置的最终结果。

稳定性分析 由图 1观察到，最终结果对初值十分敏感，在 100 组随机选取的初值条件下，求解得到的最优值差异悬殊，最大超过 0.2，最小低于 0.01，跨越了一个数量级，由此确定的最终原子位置也千差万别。其原因是在不同初值条件下，算法收敛到了不同的局部极小值，而不能收敛到全局最小值。这种情况是人们不愿意看到的，但在实际应用中却普遍存在，例如在深度学习中，模型权重的不同初始值通常会带来不同的收敛结果。为了跳出局部极小值，人们提出了多种方法，例如梯度下降中的动量项，Adam 优化器中学习率的动态调整等等。

误差分析 最终的原子位置结果如图 2和表 2所示，对应的最优值为 0.0070，这同时也是原子对距离误差的平方和，误差相对偏高。

2.1.8 结果的实际意义

在最终结果下，原子对距离误差的平方和为 0.0070，有一定的实际参考价值，但不能确定是否还有更优的解，因此该分子的结构还需进一步由实验确定。

实际上，该分子有可能是立体分子，在这种情况下，则需要更多的原子对距离约束，才能确定原子的相对位置。

2.1.9 结论

原子相对位置如图 2和表 2所示。

2.2 Chap7-Ex8 给药方案（计算题）

2.2.1 算法设计

模型 由题意，这里采用一室模型，将整个机体看作一个中心室，口服给药过程可简化为在药物进入中心室之前有一个吸收室。记中心室和吸收室的容积分别为 V 和 V_1，而 t 时刻的血药浓度分别为 $c(t)$ 和 $c_1(t)$，中心室的排除速率为 k，吸收速率为 k_1，设 $t=0$ 时刻口服剂量为 d 的药物，则吸收室的血药浓度 $c_1(t)$ 的微分方程为，

$$\frac{dc_1}{dt} = -k_1 c_1, \quad c_1(0) = \frac{d}{V_1} \tag{3}$$

中心室血药浓度 $c(t)$ 的变化率由两部分组成：与 c 成正比的排除，与 c_1 成正比的吸收。考虑到中心室和吸收室的容积分别为 V 和 V_1，得到 $c(t)$ 的微分方程为，

$$\frac{dc}{dt} = -kc + \frac{V_1}{V} k_1 c_1, \quad c(0) = 0 \tag{4}$$

由以上两个方程可解出中心室血药浓度为，

$$c(t) = \frac{d}{V}\frac{k_1}{k_1 - k}(e^{-kt} - e^{-k_1 t}) \tag{5}$$

在制定给药方案时，需要根据实验数据确定 k, k_1, b 三个参数，其中 $b = d/V$。

记给定的实验数据为 $\{(t_i, c_i)\}_{i=1}^{n}$，则确定了 n 个方程，在本题条件下 $n > 3$，因此方程组为超定方程，只能在最小二乘意义下求得最优解，其优化目标为，

$$\min z = \sum_{i=1}^{n} |c(t_i) - c_i|^2 \tag{6}$$

方程 (6) 即为本题的模型。

算法 考虑到比例系数 k, k_1 一般非负，口服剂量 d 和中心室容积 V 必然非负，即题目暗含了 $k, k_1, b \geq 0$ 这个约束，因此方程 (6) 是一个有约束优化问题，但是在实践中可以用无约束优化方法求解，只需确保最终答案满足非负约束即可，因此可以采用最速下降法，拟牛顿法的 BFGS 公式和 DFP 公式求解，对应的命令是 `fminunc`，同时，它也是一个最小二乘拟合问题，因此也可以用 LM 和置信域方法求解，对应的命令是 `lsqcurvefit`，它与 `lsqnonlin` 命令所用算法相同，但调用接口更加友好。特别的，采用置信域方法求解时，规定 k, k_1, b 的最小值均为 0。

表 3: 五种方法的 k, k_1, b 求解结果，最优值，所需迭代次数，目标函数调用次数

Method	k	k_1	b	$z_{\min}$	Iterations	Func Count
SteepDesc	/	/	/	/	/	/
BFGS	0.2803	3.6212	46.8275	34.2317	43	243
DFP	0.3228	4.4308	45.4464	103.6406	10001	40223
LM	0.2803	3.6212	46.8275	34.2317	7	35
TRM	0.2803	3.6212	46.8275	34.2317	7	32

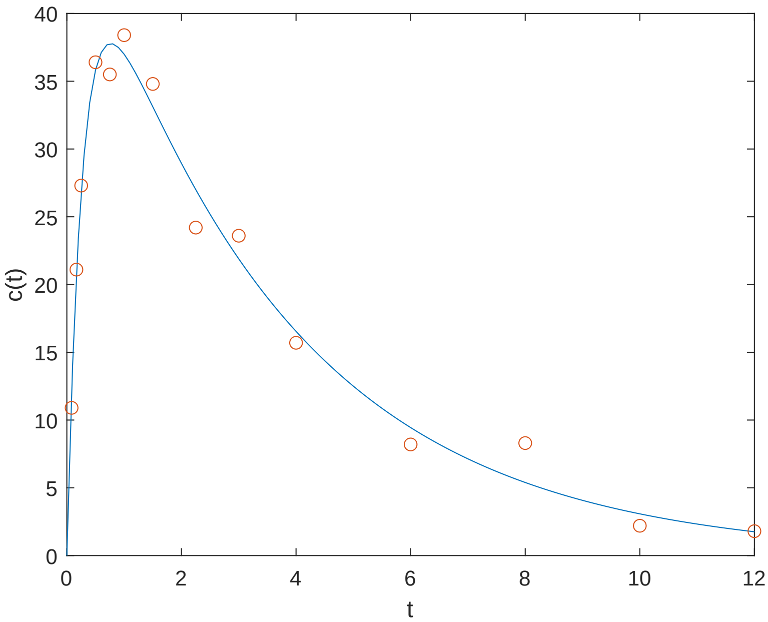

图 3: 原始实验数据以及置信域方法求得的拟合曲线

2.2.2 Matlab 程序

请参见附录 Section 4.2。

2.2.3 计算结果

在初值为 $k = 0.1, k_1 = 1, b = 10$ 时，分别用最速下降法（SteepDesc），拟牛顿法的 BFGS 和 DFP 公式，LM 方法，置信域方法（TRM）求解所得的 k, k_1, b 结果如表 3所示，其中最速下降法由于不收敛导致程序崩溃退出，求解失败。将置信域方法的求解结果作为最终结果，其对应的 $c(t)$ 拟合曲线如图 3所示。

为了进一步分析模型的稳定性，这里同样随机选取 100 组初值，令 k, k_1 在 $(0, 10)$ 区间内独立均匀地随机选取，令 b 在 $(0, 100)$ 区间内均匀地随机选取，求解得到的最优值如图 4所示，经过计算，这 100 个最优值的标准差为 4.6372×10^{-9}，k, k_1, b 的标准差分别为 4.8745×10^{-7}，4.3622×10^{-6}，3.3112×10^{-5}。

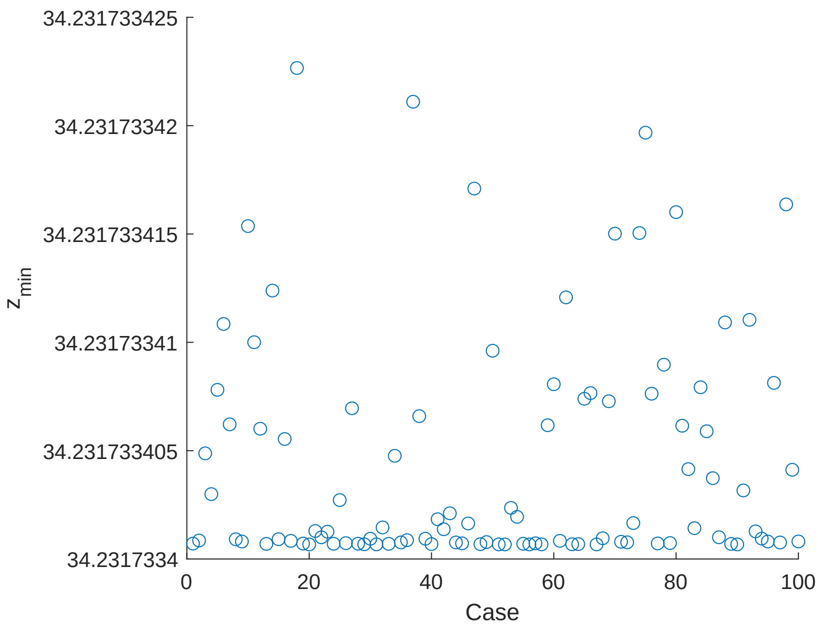

图 4: 多次随机选取初始值，采用置信域方法求解得到的最优值

2.2.4 结果分析

表 3又一次验证了，在五种方法中，最速下降法的效果最差，在本题条件下根本不收敛，拟牛顿法的 BFGS 公式和 DFP 公式的精度和效率一般，LM 方法和置信域方法的精度和效率最优。

在给定范围内多次随机选取初值，采用置信域方法求解的结果差异极小，说明模型十分稳定，对初值不敏感，而且有理由相信，表 3中的结果已经是全局最优值，因此具有实际应用价值，可作为制定给药方案的参考。

2.2.5 结论

最终求解结果为，中心室的排除速率 $k = 0.2803$，吸收速率 $k_1 = 3.6212$，口服剂量与中心室容积的比值 $b = 46.8275$。

2.3 Chap8-Ex6 投资（应用题）

2.3.1 问题分析

题目给定可供购进的有价证券的信用等级、到期年限和收益，以及银行对投资的额外限制，需要确定银行经理的投资策略。由于投资收益和所有约束条件均为关于投资金额的线性函数，因此这是一个线性规划问题。

2.3.2 模型假设

为了简化实际情况，模型基于以下假设，

1. 不存在通货膨胀和通货紧缩等货币价值变化因素。
2. 给定有价证券是零风险的，均能以给定收益率按时收回。
3. 有价证券的收益率为复利模式下折合的年收益率。

2.3.3 模型建立

第 (1) 问 将五种证券按 A,B,C,D,E 顺序排序，记第 i 种证券的信用等级为 c_i，到期年限为 t_i，到期税前收益为 p_i，银行经理购进该证券 x_i 万元，其中 $i=1,2,3,4,5$。考虑到第 2,3,4 种证券的收益需要纳税 50%，则总收益 z 最大为，

$$\max z = p_1x_1 + 0.5p_2x_2 + 0.5p_3x_3 + 0.5p_4x_4 + p_5x_5 \tag{7}$$

投资金额应当首先满足非负约束，即，

$$x_i \geq 0, \quad i=1,2,3,4,5 \tag{8}$$

题目给定的第一个投资限制可以表示为，

$$x_2 + x_3 + x_4 \geq 400 \tag{9}$$

题目给定的第二个投资限制可以表示为，

$$\frac{\sum_{i=1}^{5} c_i x_i}{\sum_{i=1}^{5} x_i} \leq 1.4 \tag{10}$$

化简为，

$$\sum_{i=1}^{5} (1.4 - c_i) x_i \geq 0 \tag{11}$$

题目给定的第三个投资限制可以表示为，

$$\frac{\sum_{i=1}^{5} t_i x_i}{\sum_{i=1}^{5} x_i} \leq 5 \tag{12}$$

化简为，

$$\sum_{i=1}^{5} (5 - t_i) x_i \geq 0 \tag{13}$$

该经理有 1000 万元资金，则需要增加预算约束，

$$\sum_{i=1}^{5} x_i \leq 1000 \tag{14}$$

综上所述，对于第 (1) 问，决策变量为 x_i ($i=1,2,3,4,5$)，目标函数为方程 (7)，约束条件为方程 (8)，方程 (9)，方程 (11)，方程 (13)，以及方程 (14)。

第 (2) 问　在第 (1) 问基础上，若该经理有 1000 万元资金，且能够以 2.75% 的利率借到不超过 100 万元资金，记实际借到 b 万元，则其取值范围约束为，

$$0 \le b \le 100 \tag{15}$$

需要修正预算约束为，

$$\sum_{i=1}^{5} x_i \le 1000 + b \tag{16}$$

将贷款利息项添加到目标函数，即，

$$\max z = p_1 x_1 + 0.5 p_2 x_2 + 0.5 p_3 x_3 + 0.5 p_4 x_4 + p_5 x_5 - 0.0275 b \tag{17}$$

综上所述，对于第 (2) 问，决策变量为 x_i $(i = 1, 2, 3, 4, 5)$ 和 b，目标函数为方程 (17)，约束条件为方程 (8)，方程 (9)，方程 (11)，方程 (13)，方程 (15)，以及方程 (16)。

第 (3) 问　同第 (1) 问。

2.3.4 算法设计

由于目标函数和约束条件均为关于决策变量的线性函数，因此这是一个线性规划问题，可采用单纯形法求解，对应的 Matlab 命令为 `linprog`，也可利用 LINGO 软件求解，并进行灵敏度分析。

2.3.5 程序

提供了 Matlab 和 LINGO 的代码，请参见附录4.3。

2.3.6 计算结果

经过实验，Matlab 的计算结果与 LINGO 完全一致，但其灵敏度分析功能较为欠缺，因此这里主要叙述更完整的 LINGO 计算结果。

第 (1) 问　LINGO 将问题的类别识别为线性规划，通过 3 次迭代计算得出，目标变量的全局最优值为 29.84，各决策变量的求解结果如表 4所示，各约束条件的约束效果如表 5所示。

表 4: 第 (1) 问各决策变量的最优值和减少费用

决策变量	x_1	x_2	x_3	x_4	x_5
最优值	218.18	0.00	736.36	0.00	45.45
减少费用	0.0000	0.0302	0.0000	0.0006	0.0000

表 5: 第 (1) 问各约束条件的松弛变量和对偶价格

约束条件	方程 (9)	方程 (11)	方程 (13)	方程 (14)
松弛变量	336.36	0.00	0.00	0.00
对偶价格	0.0000	-0.0062	-0.0024	0.0298

第 (2) 问 LINGO 同样将问题的类别识别为线性规划，通过 3 次迭代计算得出，目标变量的全局最优值为 30.07，各决策变量的求解结果如表 6所示，各约束条件的约束效果如表 7所示。

表 6: 第 (2) 问各决策变量的最优值和减少费用

决策变量	x_1	x_2	x_3	x_4	x_5	b
最优值	240.00	0.00	810.00	0.00	50.00	100.00
减少费用	0.0000	0.0302	0.0000	0.0006	0.0000	0.0000

表 7: 第 (2) 问各约束条件的松弛变量和对偶价格

约束条件	方程 (9)	方程 (11)	方程 (13)	方程 (16)	方程 (15)
松弛变量	410.00	0.00	0.00	0.00	0.00
对偶价格	0.0000	-0.0061	-0.0024	0.0298	0.0023

第 (3) 问 通过 LINGO 的 Prices & Ranges 功能分析得出模型的灵敏度，即当最优基矩阵不变时，各决策变量对应系数的变化范围，如表 8所示。可以看出，若证券 A 的税前收益增加为 4.5%，即增加 0.0020 时，增加幅度低于其允许增加的最大幅度 0.0035，因此投资策略不应改变，经过计算得出，此时的收益为 30.27 万元；若证券 C 的税前收益减少为 4.8%，即减少 0.0020 时，减少幅度高于其允许减少的最大幅度 0.0006，投资应该改变，经过计算，此时的最优解如表 9所示，收益最大为 29.42 万元。

表 8: 第 (3) 问模型的敏感性分析

决策变量	x_1	x_2	x_3	x_4	x_5
当前系数	0.0430	0.0270	0.0250	0.0220	0.0450
允许增加	0.0035	0.0302	0.0173	0.0006	0.0520
允许减少	0.0130	∞	0.0006	∞	0.0140

表 9: 当证券 C 的税前收益减少为 4.8% 时，各决策变量的最优值及减少费用

决策变量	x_1	x_2	x_3	x_4	x_5
最优值	336.00	0.00	0.00	648.00	16.00
减少费用	0.0000	0.0306	0.0004	0.0000	0.0000

2.3.7 结果的数学分析

在第 (1) 问中，(x_1, x_3, x_5) 的减少费用为零，是一个最优基矩阵，方程 (11)，方程 (13)，方程 (14) 的松弛变量为零，起到约束作用。

在第 (2) 问中，(x_1, x_3, x_5) 同样是一个最优基矩阵，方程 (11)，方程 (13)，方程 (15)，方程 (16) 的松弛变量为零，起到约束作用。此外，从对偶价格的角度来看，由于第 (1) 问中方程 (14) 的对偶价格为 0.0298，高于贷款利率 2.75%，并且灵敏度分析表明其允许增长幅度为 ∞，因此，经理应该借尽可能多的钱来买证券。

2.3.8 结果的实际意义

这里的计算结果对制定投资策略具有一定的参考意义，但本模型相对简单，未考虑诸多现实因素。在实际情况下，还需综合考虑通货膨胀等货币贬值因素，根据有价证券的信用等级和到期年限进行风险评估，分散投资风险，并且切勿盲目借钱投资，做出明智的决策。

2.3.9 结论

1. 若该经理有 1000 万元资金，应当购进 A 证券 218.18 万元，购进 C 证券 736.36 万元，购进 E 证券 45.45 万元，不购进其他证券，此时收益最大，为 29.84 万元。

2. 如果能够以 2.75% 的利率借到不超过 100 万元资金，该经理应借入 100 万元，购进 A 证券 240.00 万元，购进 C 证券 810.00 万元，购进 E 证券 50.00 万元，不购进其他证券，此时收益最大，为 30.07 万元。

3. 在 1000 万元资金情况下，若证券 A 的税前收益增加为 4.5%，则不应改变投资策略，此时最大收益为 30.27 万元；若证券 C 的税前收益减少为 4.8%，则应改变投资策略为，购进 A 证券 336.00 万元，购进 D 证券 648.00 万元，购进 E 证券 16.00 万元，不购进其他证券，此时收益最大，为 29.42 万元。

3 收获与建议

在本次实验中，我掌握了 Matlab 优化工具箱和 LINGO 软件的基本用法，对不同算法进行了初步分析和比较，用无约束优化方法以及线性规划方法建立了实际问题的模型，并进行求解，在解决实际问题的过程中，我对数学方法的原理和应用有了更深刻的理解。

希望助教能对每次的实验进行详细的解答，希望老师在未来的课堂上介绍更多数学应用的前沿知识。

4 附录：程序代码

4.1 Chap7-Ex5

```
dist = [
    4 1 0.9607; 5 4 0.4758; 18 8 0.8363; 15 13 0.5725;
    12 1 0.4399; 12 4 1.3402; 13 9 0.3208; 19 13 0.7660;
    13 1 0.8143; 24 4 0.7006; 15 9 0.1574; 15 14 0.4394;
    17 1 1.3765; 8 6 0.4945; 22 9 1.2736; 16 14 1.0952;
    21 1 1.2722; 13 6 1.0559; 11 10 0.5781; 20 16 1.0422;
    5 2 0.5294; 19 6 0.6810; 13 10 0.9254; 23 16 1.8255;
    16 2 0.6144; 25 6 0.3587; 19 10 0.6401; 18 17 1.4325;
    17 2 0.3766; 8 7 0.3351; 20 10 0.2467; 19 17 1.0851;
    25 2 0.6893; 14 7 0.2878; 22 10 0.4727; 20 19 0.4995;
    5 3 0.9488; 16 7 1.1346; 18 11 1.3840; 23 19 1.2277;
    20 3 0.8000; 20 7 0.3870; 25 11 0.4366; 24 19 1.1271;
    21 3 1.1090; 21 7 0.7511; 15 12 1.0307; 23 21 0.7060;
    24 3 1.1432; 14 8 0.4439; 17 12 1.3904; 23 22 0.8052;
]';

coords0 = zeros(48,1);

% options = optimoptions('fminunc', 'Algorithm', 'quasi-newton', ...
%     'HessUpdate', 'bfgs', 'MaxFunEvals', 1000000, 'MaxIter', 10000)
    ;
% [coords, fval, exitflag, output] = fminunc(@fun, coords0, options,
    dist);

options = optimoptions('lsqnonlin', 'Algorithm', 'levenberg-marquardt
    ',...
    'MaxFunEvals', 1000000, 'MaxIter', 10000);
[coords,resnorm,residual,exitflag,output] = ...
    lsqnonlin(@lsqfun, coords0, [], [], options, dist);

```

```
x = [coords(1:2:48); 0];
y = [coords(2:2:48); 0];
figure; scatter(x, y); xlabel('x'); ylabel('y'); grid on;
text(x, y+0.1, num2str((1:length(x))'), 'HorizontalAlignment', 'center');

function [x, y] = get_coord(coords, index)
    if index == length(coords) / 2 + 1
        x = 0;
        y = 0;
    else
        x = coords(2 * index - 1);
        y = coords(2 * index);
    end
end

function err = fun(coords, dist)
    err = 0;
    for pair = dist
        d_ij = pair(3);
        [x_i, y_i] = get_coord(coords, pair(1));
        [x_j, y_j] = get_coord(coords, pair(2));
        err = err + (sqrt((x_i - x_j)^2 + (y_i - y_j)^2) - d_ij)^2;
    end
end

function y = lsqfun(coords, dist)
    y = zeros(size(dist, 2), 1);
    for i = 1:length(y)
        pair = dist(:, i);
        d_ij = pair(3);
        [x_i, y_i] = get_coord(coords, pair(1));
        [x_j, y_j] = get_coord(coords, pair(2));
        y(i) = sqrt((x_i - x_j)^2 + (y_i - y_j)^2) - d_ij;
    end
end
```

4.2 Chap7-Ex8

```
t = [0.083 0.167 0.25 0.50 0.75 1.0 1.5 2.25 3.0 4.0 6.0 8.0 10.0 12.0]';
c = [10.9 21.1 27.3 36.4 35.5 38.4 34.8 24.2 23.6 15.7 8.2 8.3 2.2 1.8]';

```

```
x0 = [0.1 1 10]';

% options = optimoptions('fminunc', 'Algorithm', 'quasi-newton', ...
%     'HessUpdate', 'bfgs', 'MaxFunEvals', 1000000, 'MaxIter', 10000);
% [x, fval, exitflag, output] = fminunc(@fun, x0, options, t, c);

options = optimoptions('lsqcurvefit', 'Algorithm', 'levenberg-marquardt',...
    'MaxFunEvals', 1000000, 'MaxIter', 10000);
[x,resnorm,residual,exitflag,output] = ...
    lsqcurvefit(@lsqfun, x0, t, c, [], [], options);

k = x(1); k1 = x(2); b = x(3);
t_plot = 0:0.1:12;

figure; plot(t_plot, concentration(k, k1, b, t_plot));
hold on; scatter(t, c); xlabel('t'); ylabel('c(t)');

function c = concentration(k, k1, b, t)
    c = b * k1 / (k1 - k) * (exp(-k*t) - exp(-k1*t));
end

function err = fun(x, t, c)
    k = x(1); k1 = x(2); b = x(3);
    err = sum((concentration(k, k1, b, t) - c).^2);
end

function c = lsqfun(x, t)
    k = x(1); k1 = x(2); b = x(3);
    c = concentration(k, k1, b, t);
end
```

4.3 Chap8-Ex6

4.3.1 Matlab

```
credit = [2 2 1 1 5]';
due = [9 15 4 3 2]';
profit = [0.043 0.054 0.050 0.044 0.045]';
tax = [0 0.5 0.5 0.5 0]';

f = (1-tax) .* profit;

```

```
A = [0 -1 -1 -1 0;
    credit' - 1.4;
    due' - 5;
    ones(1,5)];
b = [-400 0 0 1000]';

[x,fval,exitflag,output,lambda] = ...
    linprog(-f, A, b, [], [], zeros(5,1), []);
x, fval

f = [f; -0.0275];
A = [A [0 0 0 -1]'];
[x,fval,exitflag,output,lambda] = ...
    linprog(-f, A, b, [], [], zeros(6,1), [inf*ones(5,1); 100]);
x, fval
```

4.3.2 LINGO

第 (1) 问和第 (3) 问的代码如下，

```
model:

sets:
bond/1..5/: credit, due, profit, tax, x;
endsets

data:
credit = 2 2 1 1 5;
due = 9 15 4 3 2;
profit = 0.043 0.054 0.050 0.044 0.045;
tax = 0 0.5 0.5 0.5 0;
enddata

max = @sum(bond: (1 - tax) * profit * x);

x(2) + x(3) + x(4) >= 400;
@sum(bond: (1.4 - credit) * x) >= 0;
@sum(bond: (5 - due) * x) >= 0;

@sum(bond: x) <= 1000;

end
```

第 (2) 问代码如下，

```
model:
```

```

sets:
bond/1..5/: credit, due, profit, tax, x;
endsets

data:
credit = 2 2 1 1 5;
due = 9 15 4 3 2;
profit = 0.043 0.054 0.050 0.044 0.045;
tax = 0 0.5 0.5 0.5 0;
interest_rate = 0.0275;
enddata

max = @sum(bond: (1 - tax) * profit * x) - interest_rate * borrow;

x(2) + x(3) + x(4) >= 400;
@sum(bond: (1.4 - credit) * x) >= 0;
@sum(bond: (5 - due) * x) >= 0;

@sum(bond: x) <= 1000 + borrow;
borrow < 100;

end
```