

编号

南京航空航天大学

毕业论文

题目 基于图模型理论的冲突敏感性分析系统开发

学生姓名

学号

学院

理学院

专业

信息与计算机科学

班级

指导教师

二〇一六年六月

南京航空航天大学

本科毕业设计（论文）诚信承诺书

本人郑重声明：所呈交的毕业设计（论文）（题目：基于图模型理论的冲突敏感性分析系统开发）是本人在导师的指导下独立进行研究所取得的成果。尽本人所知，除了毕业设计（论文）中特别加以标注引用的内容外，本毕业设计（论文）不包含任何其他个人或集体已经发表或撰写的成果作品。

作者签名：

年 月 日

（学号）：

基于图模型理论的冲突敏感性分析系统开发

摘要

本文设计开发的冲突敏感性分析系统，是基于图模型理论展开的。

论文以两个决策者的冲突分析为切入点，首先介绍了基于图模型理论的四种稳定性的定义；其次，从四种稳定性的定义出发，构建相应代数表达式；接下来，在稳定性代数表达式的基础上展开冲突敏感性条件的探讨；然后在稳定性及敏感性代数表达式的探究结果基础上运用 MATLAB GUI 进行敏感性分析系统的构建；最后，运用冲突敏感性分析系统对一个军事案例进行了分析。

关键词：冲突分析，冲突分析敏感性，两个决策者，严格有序偏好，系统开发

The development of conflict sensitivity analysis system based on graph model theory

Abstract

In this paper, design and development of the conflict sensitivity analysis system is based on the theory of graph model.

The conflict with the two decision analysis as a starting point, first ,introduced based on the definition of graph theory model of four kinds of stability; secondly, from the definition of four kinds of stability of construct the corresponding algebraic expression; next, on the stability of algebraic expressions based on the expansion process sensitivity of; then in stability and sensitivity of algebraic expression for the inquiry on the basis of the result using Matlab GUI to construct the sensitivity analysis of the system; finally, use of conflict sensitivity analysis system has carried on the analysis to the case of a military.

Key words: Conflict analysis; Sensitivity of conflict analysis; Two decision makers; Strictly ordered preference; System development.

目录

摘要..... i

Abstract..... ii

第三章 冲突敏感性分析系统开发..... - 1 -

 3.1 系统开发环境及工具介绍..... - 1 -

 3.2 冲突敏感性分析系统构建思路与方法..... - 2 -

 3.3 冲突稳定性主要算法及系统实现..... - 3 -

 3.3.1 纳什（Nash）稳定..... - 3 -

 3.3.2 一般超理性（GMR）稳定..... - 4 -

 3.3.3 对称超理性（SMR）稳定..... - 5 -

 3.3.4 序列（SEQ）稳定..... - 6 -

 3.4 冲突敏感性主要算法及系统实现..... - 7 -

 3.5 参数传递与模块互联..... - 9 -

 3.6 异常处理..... - 10 -

 3.7 本章小结..... - 12 -

第四章 冲突敏感性案例分析..... - 13 -

 4.1 冲突背景描述..... - 13 -

 4.2 冲图建模与系统验证..... - 13 -

 4.2.1 冲突分析..... - 13 -

 4.2.2 冲突分析敏感性..... - 15 -

 4.3 案例拓展..... - 17 -

 4.4 本章小结..... - 18 -

附录..... - 19 -

【理论部分已删除】

第三章 冲突敏感性分析系统开发

3.1 系统开发环境及工具介绍

MATLAB 对于理工科学生来说，是再熟悉不过的软件了，它是 MATrix LABoratory（矩阵实验室）的缩写，由美国 The MathWorks 公司开发的一款商业数学软件。是一种主要用于算法开发、数据分析、数据可视化以及数值计算的高级技术计算语言和交互式环境。除了矩阵运算、绘制函数/数据图像等常用功能外，MATLAB 还可以用来创建用户界面及与调用其它语言（如 C，C++，FORTRAN 等）编写的程序。

而在本毕业设计冲突敏感性分析系统的构建开发过程中，除用到 MATLAB 基本的数值计算、算法开发等功能外，还着重使用 MATLAB GUI 构建图形用户界面，使得即使对 MATLAB 一无所知的用户也能轻松操作该系统，使本系统真正做到能为广大有需要的普通所用。

那么，MATLAB GUI 究竟是什么呢？正如 MATLAB 是矩阵实验室 MATrix LABoratory 的缩写一样，GUI 的全称是 Graphical User Interface，即图形用户界面。和许多高级编程语言一样，MATLAB 也有图形用户界面开发环境。也许我们会发现，随着计算机技术的飞速发展，人与计算机的通信方式也发生的很大的变化，从原来的命令行通讯方式（如 DOS 系统、Linux 系统等）变化到了现在的图形界面下的交互方式，而现在绝大多数的应用程序都是在图形化用户界面下运行的。由此可见，GUI 就相当于 MATLAB 中的 Visual Basic，是面向对象的编程，可以使所有终端用户更易操作该程序！换句话说，如果没有 GUI，我们只能比较繁琐的从 Command Window(命令行)中调用相关程序，并且不是所有用户都是这个能力。

创建 MATLAB GUI 界面通常有两种方式：方式一，使用 .m 文件直接进行动态控件的添加；方式二，使用 GUIDE 快速的生成 GUI 界面。因为方式二创建 GUI 的方法更加方便快捷，适合大型程序的可视化编辑，固本毕业设计选择方法二来进行系统开发。

3.2 冲突敏感性分析系统构建思路与方法

本毕设所构建的冲突敏感性分析系统分为状态点稳定类型判断及平衡点计算（即稳定性分析）和敏感性分析两个模块。

稳定性分析模块以四种稳定性的代数表达式为理论依据，通过 MATLAB 编写相应算法实现各状态点的稳定性类型判断，并在可视化界面进行反馈输出。敏感性分析模块通过更直观、易于算法实现的代数判定条件将原本抽象的、难以通过算法实现的复杂判断简单化，使得通过算法进行敏感性分析变为可能。

在实际系统开发构建中，我们首先从状态转移矩阵和严格有序偏好序列出发，结合稳定性代数判定式进行各状态点稳定类型及平衡点的计算输出；其次，选择某一平衡点，输入互换状态点、稳定类型，结合敏感性代数判别条件进行敏感性分析；最后，给出敏感性判定结果，对实际决策发挥指导意义。

是以，本系统需要输入的数据包括以下几项：

- （1）状态点与状态转移矩阵；
- （2）决策者信息；
- （3）初始有序偏好；
- （4）用于敏感性分析的稳定点；
- （5）稳定点敏感性分析类型；
- （6）进行敏感性分析互换的状态点。

需要输出的信息包括：

- （1）各类稳定性对应的状态点；
- （2）平衡点；
- （3）偏好发生变化后新的偏好序列；
- （4）偏好改变后平衡点稳定性的变化。

综上，系统 GUI 界面设计如下图所示：

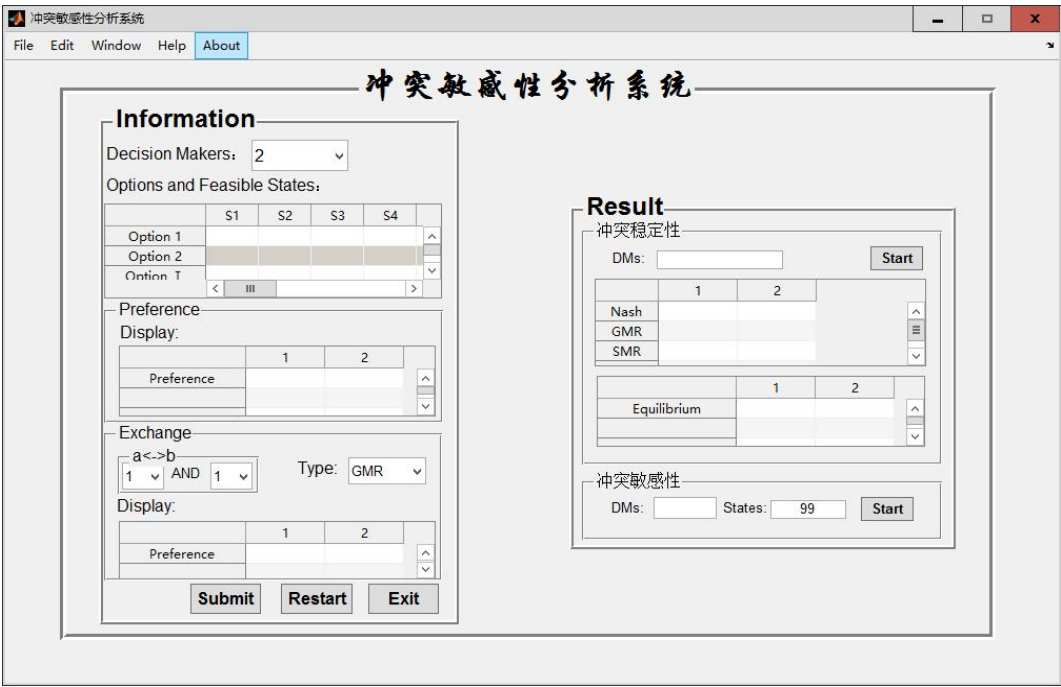


图 3.1 系统 GUI 界面设计

左侧 Information 块主要用于信息的输入与展示，右侧 Result 块用于结果的展示输出，其中，偏好序列的输入和偏好改变后平衡点稳定性的变化等将通过 dialog 对话框进行实现。

3.3 冲突稳定性主要算法及系统实现

在上一节我们已经根据需要输入输出的信息进行的冲突敏感性分析系统图形用户界面的设计，在本节及接下来的一节内容中，我们将着重对状态点稳定性类型及冲突敏感性分析的算法实现进行论述。

注：在以下稳定性及敏感性的讨论中，我们以决策者 i 为例。

3.3.1 纳什（Nash）稳定

由定义 2.6 及定理 1 可知，当状态点 s 纳什稳定时，必有 $J_i^+(s,:) = \text{zeros}(1,m)$ ，即对决策者 i 来说，从状态点 s 出发的所有可达状态点中，不存在比 s 更优的点，所以算法如下：

```
i=1;
Nashi=[];                                     %用于保存所有纳什稳定的状态点
for s=1:m1                                     %m1 为  $J_i^+$  的行数
```



```

if J_ii(s,:)==zeros(1,n1)      %n1 为  $J_i^+$  的列数
    Nashi(i)=s;
    i=i+1;
    continue
end
end
Nashi;

```

3.3.2 一般超理性（GMR）稳定

由定理 2 和定义 2.7 可知，一般超理性的冲要条件是 $J_i^+(s,h) \bullet V_i^{GMR}(h) = 0$ ，其中 $V_i^{GMR} = 1 - \max_{1 \leq q \leq m} [J_j(h,q) \bullet (P_i(s) - P_i(q))]$ ，因 3.3.1 已讨论 Nash 稳定的情况，根据四种稳定性的包含关系可知，无需再对 Nash 稳定点进行讨论。固当改良可达矩阵对应元素为 1 且 $(J_j(h,q) \bullet (P_i(s) - P_i(q))) > 0$ 时，为 GMR 稳定。我们可以用图 3.2 来说明这种稳定（其中 q^- 表示对于决策者 i 来说， $q \prec s$ ； q^+ 表示 $q \succ s$ 。下同）：

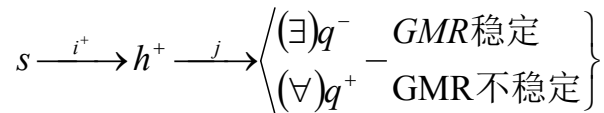


图 3.2 GMR 稳定图

所以我们可得如下算法程序：

```

i=1;
GMRi=[];
for s=1:m1
    if J_ii(s,:)==zeros(1,n1) %此部分是 Nash 稳定点，已输出，固 continue
        continue
    end
    for h=1:m2      %m2 为决策者 j 可达矩阵的行
        if J_ii(s,h)==0
            continue
        end
        for q=1:n2  %n2 为决策者 j 可达矩阵的列
            a=J_j(h,q)*(P_i(1,s)-P_i(1,q));
            if a>0
                GMRi(i)=s;
            end
        end
    end
end

```

```

        i=i+1;
        break
    end
end
end
end
end
GMRi;

```

3.3.3 对称超理性（SMR）稳定

根据定义 2.8 和定理 3 的论述，我们把对称超理性稳定用下图直观的进行表示：

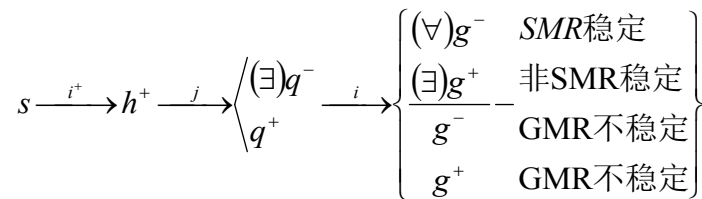


图 3.3 SMR 稳定图

与上一小节类似，我们有如下算法程序：

```

i=1;
SMRi=[];
for s=1:m1
    if J_ii(s,:)==zeros(1,n1) %此部分是 Nash 稳定点，已输出，固 continue
        continue
    end
    for h=1:m2 %m2 为决策者 j 可达矩阵的行
        if J_ii(s,h)==0
            continue
        end
        for q=1:n2
            a=J_j(h,q)*(P_i(1,s)-P_i(1,q));
            if a>0

                for g=1:n1
                    k=0;
                    a=J_j(q,g)*(P_i(1,s)-P_i(1,g));
                    if a>0 %证明 GMR 稳定
                        k=k+1;
                    end
                end
            end
            if k==n1 %表明对于所有 g 都有(P_i(1,s)>P_i(1,g),即 s 点更优
                SMRi(i)=s;
            end
        end
    end
end

```

```

        i=i+1;
    end
end
end
end
end
SMRi;

```

3.3.4 序列（SEQ）稳定

很明显，对比定理 2 和定理 4，我们发现序列稳定和一般超理性稳定唯一的不同点是决策者 j 从状态点 $s1$ 出发是可达还是改良可达，所以只需将 GMR 算法程序第三个 for 语句中的 $a=J_j(h, q)*(P_i(1, s)-P_i(1, q))$; 改为 $a=J_jj(h, q)*(P_i(1, s)-P_i(1, q))$; 即可。

```

i=1;
SEQi=[];
for s=1:m1
    if J_ii(s,:)==zeros(1,n1) %此部分是 Nash 稳定点，已输出，固 continue
        continue
    end
    for h=1:m2
        if J_ii(s,h)==0
            continue
        end
        for q=1:n2
            a=J_jj(h,q)*(P_i(1,s)-P_i(1,q));
            if a>0
                SEQi(i)=s;
                i=i+1;
                break
            end
        end
    end
end
end
SEQi;

```

经过以上稳定性分析，我们已经可以得到各稳定类型所对应的状态点及平衡点了，在下一节，我们将对冲突敏感性主要算法及系统实现展开讨论。

3.4 冲突敏感性主要算法及系统实现

通过上一节的探究，我们根据代数表达式以及输入的状态转移矩阵、偏好信息得出了各状态点的稳定类型，这一节，我们依旧以决策者 i 为例，在稳定性的基础上，展开对敏感性的探究。

由于篇幅所限，在本节中，我们以 GMR 稳定为例进行敏感性分析，并在本节讨论结束时附上相应代码，以供参详。

由第二章敏感性分析的理论部分，我们总结出进行 GMR 冲突敏感性分析的流程图如下：

【已删除】

由流程图可知，我们需要输入互换的状态点，需要定义并生成一个用于敏感性判断的集合 $w(s_1)$ ，即定义 2.11 中的 $w(s_1)$ 。

获取用于交换的状态点及 $w(s_1)$ 的算法代码如下：

```
sa=get(handles.popupmenu10,'string');    %交换的点
sb=get(handles.popupmenu12,'string');
sa=str2num(sa);
sb=str2num(sb);

T=J_ii(s,:);
T(:,sa)=[];    %去掉 sa 列
[x]=find(T(1,:));    %提取非零元素

[m,n]=size(x);    %m=1, n 为非零元素个数
[r,t]=size(J_ii);
k=1;
w=[];
for i=1:n
    s1=x(i);
    for s2=1:r
        if (J(s1,s2)*(P_i(s)-P_i(s2)))>0
            w(k)=s2;
            k=k+1;
        end
    end
end
if i==n
    w;
    [m,n]=size(w);    %n 为 w 中元素的个数
```

end
End

得出 $w(s_1)$ 即可根据流程图所示进行敏感性算法的开发，代码如下：

```
if (P_i(sa)>P_i(s) & P_i(sb)>P_i(s)) | (P_i(sa)<P_i(s) & P_i(sb)<P_i(s))    %优先级同侧交换
    disp(sprintf('状态点 %d 对决策者 i（中国）保持 %s 稳定性不变(type1)!',s,type))
    elseif J_i(s,sb)==0
        if (n>=2 | (n==1 & w~=sb) | (n==1 & w==sb &
J(s1,sa)==1))                                %优先级异侧交换
            disp(sprintf('状态点 %d 对决策者 i（中国）保持 %s 稳定性不变(type2)!',s,type))
            else
                disp(sprintf('状态点 %d 对决策者 i（中国） %s 稳定性改变(type2)!',s,type))
            end
        elseif J_i(s,sb)==1

            T=J_ii(s,:);
            T(:,sa)=[];                                %去掉 sa 列
            [x]=find(T(1,:));                            %提取非零元素
            x=union(x,sb);                                %求并

            [m,n]=size(x);                                %m=1, n 为非零元素个数
            [r,t]=size(J_ii);
            k=1;
            w=[];
            for i=1:n
                s1=x(i);
                for s2=1:r
                    if (J(s1,s2)*(P_i(s)-P_i(s2)))>0
                        w(k)=s2;
                        k=k+1;
                    end
                end
            end
            if i==n
                w;
                [m,n]=size(w);                            %n 为 w 中元素的个数
            end
        end
        if (n>=2 | (n==1 & w~=sb) | (n==1 & w==sb & J(s1,sa)==1))
            disp(sprintf('状态点 %d 对决策者 i（中国）保持 %s 稳定性不变(type3)!',s,type))
        else
            disp(sprintf('状态点 %d 对决策者 i（中国） %s 稳定性改变(type3)!',s,type))
        end
    end
end
```

end

End

至此，我们冲突敏感性分析系统主要功能模块已基本实现，但是我们在可视化界面上输入的信息如何为程序所用仍然没有得到解决，所以说这样的系统虽然看似实现了我们想要的功能，实则无法运行使用，对于终端用户来说毫无价值。在接下来的一节中，我们将对 MATLAB 句柄函数进行介绍和使用，使得系统可以真正“跑”起来。

3.5 参数传递与模块互联

通过前面几节的探究，我们已经得出了实现稳定性和敏感性判断的算法程序代码，而要是系统能够“跑”起来，却也离不开强大而繁琐的句柄函数。

首先，我们来认识两个重要的函数：set 和 get 函数。

我们要运行系统时，首先会点击 Result 中的“冲突稳定性”栏 start，此时会检测决策者信息栏（DMS）中信息的正确性，如果有误则报错（异常处理）并弹出对话框重新输入。并将重新输入的值赋给 DMS 栏，检查无误后执行后面的程序。这里用到了

`ch=get(handles.edit5,'string');`和`set(handles.edit5,'string',ch);`分别表示从项目 edit5（即 DMS）获得值或给 edit5 设置一个值。有了这两个函数，我们只需将模块间进行传递的值放到项目里即可方便的进行转换和传递，同时，稳定性分析和敏感性分析的结果输出也不再是问题。

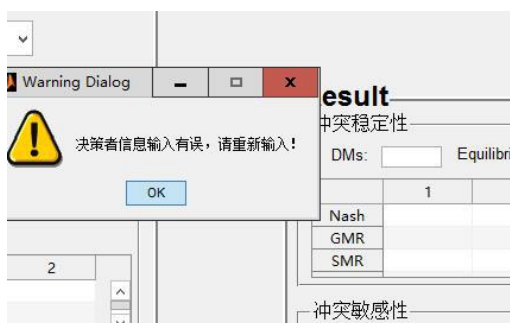


图 3.5 信息异常提示

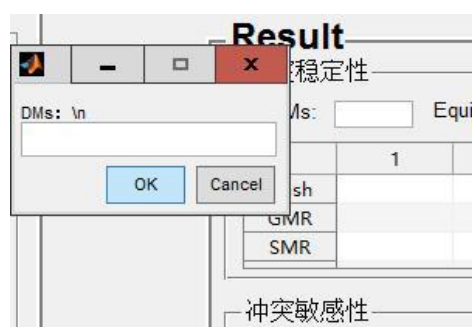


图 3.6 信息输入

其次，通过对话框输入偏好信息后，由于数据类型不同，如何将 cell 类型的数据进行转换，存储到矩阵中。

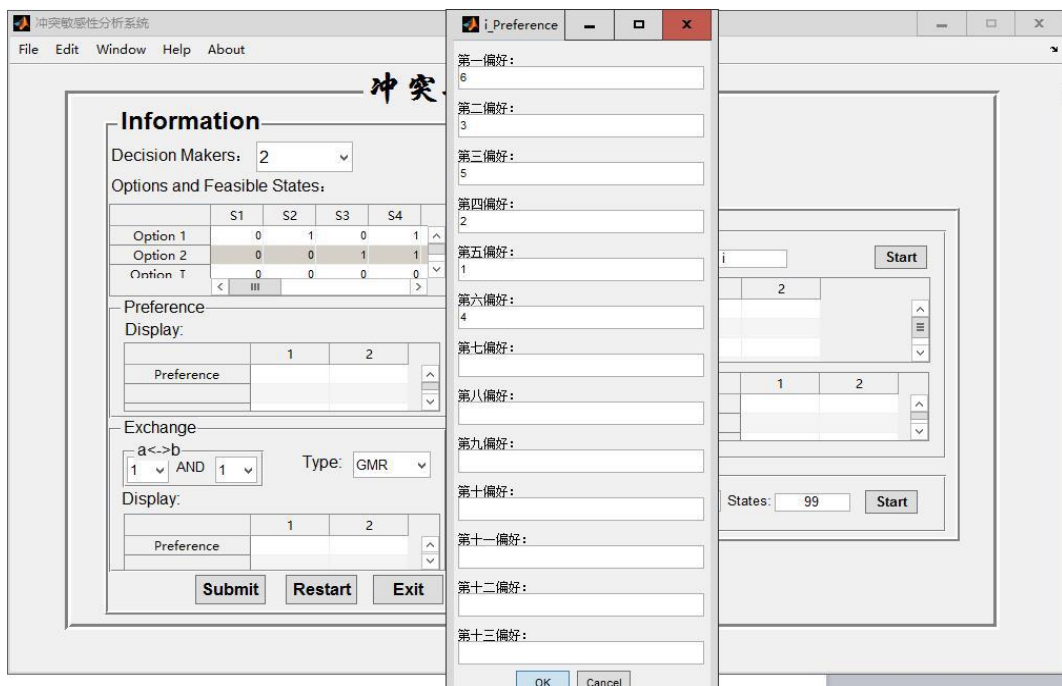


图 3.7 偏好输入与转换

转化代码如下：

```
prompt={'第一偏好: ','第二偏好: ','第三偏好: ','第四偏好: ','第五偏好: ','第六偏好: ','第
七偏好: ','第八偏好: ','第九偏好: ','第十偏好: ','第十一偏好: ','第十二偏好: ','第
十三偏好: '}; %设置提示字符串, 输入偏好序列
name='i_Preference'; %设置标题
numlines=1; %指定输入数据的行数
def={'6','3','5','2','1','4',' ',' ',' ',' ',' '}; %设定默认值
Resize='on'; %设定对话框尺寸可调节
value_p=inputdlg(prompt,name,numlines,def,'on'); %创建输入对话框
A=[str2num(value_p{1}) str2num(value_p{2}) str2num(value_p{3})...
str2num(value_p{4}) str2num(value_p{5}) str2num(value_p{6})...
str2num(value_p{7}) str2num(value_p{8}) str2num(value_p{9}) ...
str2num(value_p{10}) str2num(value_p{11}) str2num(value_p{12}) ...
str2num(value_p{13})]; % value_p 为 'cell'型, 用 str2num(value_p{ })获取和转换
```

当然，MATLAB 句柄函数在 GUI 界面开发中运用广泛，作用巨大，在此只是对本系统中用到的部分函数进行说明，未能足够详实。

3.6 异常处理

在本系统的输入、计算、输出中，难免会因为使用者粗心或者使用不当造成这样那样的错误，一个好的系统应该能够对这些错误进行规避，或者是给用户必要的提示。本系统也使

用了必要的异常处理机制。

DMS 信息输入有误时，跳出如下图所示的警告对话框，并要求使用者重新输入。当输入了一个正确的决策者时，提醒用户所输入的决策者信息。



图 3.8 错误警告对话框



图 3.9 信息提示对话框

相关代码：

```
k=1;
while(k==1)
    ch=get(handles.edit5,'string');
    if ch=='i'
        h=msgbox('您输入的决策者为 i（中国）！','DMs');
        while(ishandle(h))           %等待对话框被关闭,再执行后面程序
            pause(1) %当对话框未被关闭，延迟 1 秒再执行下一 while 循环，防止 CPU 占用太大
        end

        k=0;
    elseif ch=='j'
        h=msgbox('您输入的决策者为 j（欧盟）！','DMs');
        while(ishandle(h))           %等待对话框被关闭,再执行后面程序
            pause(1) %当对话框未被关闭，延迟 1 秒再执行下一 while 循环，防止 CPU 占用太大
        end
        k=0;
    else

        h = warndlg('决策者信息输入有误，请重新输入！');
        while(ishandle(h))           %等待对话框被关闭,再执行后面程序
            pause(1) %当对话框未被关闭，延迟 1 秒再执行下一 while 循环，防止 CPU 占用太大
        end

        ch=inputdlg('DMs: \n');
        ch=ch{1};
        set(handles.edit5,'string',ch) %把接收的值赋给 DMS 栏
        k=1;
    end
end
End
```


当进行敏感性分析的状态点不存在时：



图 3.10 提示状态点不存在

提示确定敏感性分析类型：

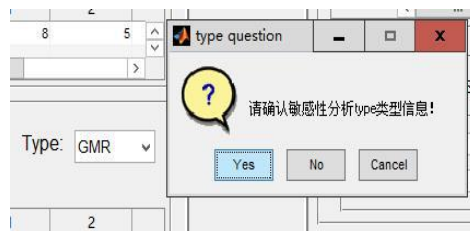


图 3.11 提示信息确认

当输入用于交换的状态点下标越界时：

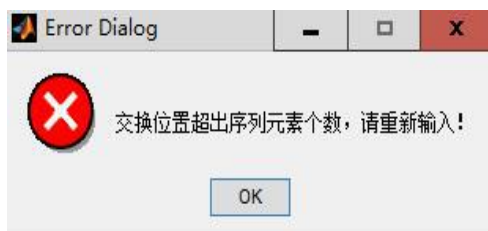


图 3.12 下标越界提示

提示敏感性分析结果：



图 3.13 结果展示

当敏感性分析点与分析类型不匹配：



图 3.14 分析类型不匹配

敏感性分析完成，提示是否清除全局变量：

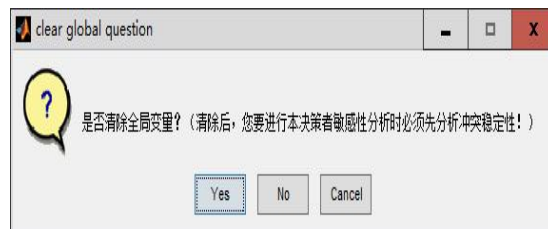


图 3.15 提示并警示

本系统通过一系列对话框对用户的使用进行引导和提醒，使得视同运行更加稳定可靠，保证了广大普通用户能够正常使用。

3.7 本章小结

本章首先对 MATLAB GUI 进行了一个初步的介绍，使普通使用者对其能有一个大概的了解；其次将系统构建分为两个模块，进行功能剖析，得出 GUI 界面中必备的相关要素；然后以第二章相关理论为依据，由相应代数表达式构建相关算法，用于实现系统稳定性分析与敏感性分析的主要功能；接着又通过对相关句柄函数的介绍和使用，使系统联系为一个可以运行的整体；最后，通过异常处理和提示提高系统稳定性，增强用户体验，最终实现系统功能。

第四章 冲突敏感性案例分析

4.1 冲突背景描述

若 A、B 两个国家（决策者 1 和决策者 2）之间可能爆发一场军事冲突，已知国家 A 可选的行动方案（Option）有从空中进攻（A-1）、从海上进攻（A-2）以及进行和谈（A-3）。B 国可选的行动方案有空中防御（B-1）、海上防御（B-2）和和平谈判（B-3）。假设双方对对方实际可用的方案和策略都有了正确的认知，方案和冲突可能的结局列于表 4.1 所示。

表 4.1 方案和冲突可能的结局

方 案	结局编号												
	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13
A-1	N	Y	Y	Y	Y	N	N	N	N	Y	Y	Y	Y
A-2	N	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y
A-3	Y	N	N	N	N	N	N	N	N	N	N	N	N
B-1	N	Y	N	N	Y	Y	N	N	Y	Y	N	N	Y
B-2	N	N	Y	N	Y	N	Y	N	Y	N	Y	N	Y
B-3	Y	N	N	Y	N	N	N	Y	N	N	N	Y	N

A、B 两国的真实偏好向量列于表 4.2。

表 4.2 A、B 两国真实偏好

项目	符号	偏好向量(权值)
A 的真实偏好向量	V1	(3, 4, 10, 13, 6, 9, 2, 12, 5, 7, 8, 11, 1)
B 的真实偏好向量	V2	(13, 11, 5, 2, 9, 7, 12, 3, 10, 6, 4, 1, 8)

4.2 冲突建模与系统验证

4.2.1 冲突分析

在对手决策者策略不变的条件下单方面改变己方行动策略，由表 4.1，我们可分别得出 A 国和 B 国的状态转移矩阵，分别以 J_i 和 J_j 表示，如下图所示：

$J_i =$

0	0	0	1	0	0	0	1	0	0	0	1	0
0	0	0	0	0	1	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1	0	0
1	0	0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	0	0	0	0	1	0	0	0	1
0	1	0	0	0	0	0	0	0	1	0	0	0
0	0	1	0	0	0	0	0	0	0	1	0	0
1	0	0	1	0	0	0	0	0	0	0	1	0
0	0	0	0	1	0	0	0	0	0	0	0	1
0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0
1	0	0	1	0	0	0	1	0	0	0	0	0
0	0	0	0	1	0	0	0	1	0	0	0	0

图 4.1 A 国状态转移矩阵 J_i

$J_j =$

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0	0	0	0
0	1	0	1	1	0	0	0	0	0	0	0	0
0	1	1	0	1	0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	0	0	0	0
0	0	0	0	0	1	0	1	1	0	0	0	0
0	0	0	0	0	1	1	0	1	0	0	0	0
0	0	0	0	0	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	1	1
0	0	0	0	0	0	0	0	0	1	0	1	1
0	0	0	0	0	0	0	0	0	1	1	0	1
0	0	0	0	0	0	0	0	0	1	1	1	0

图 4.2 B 国状态转移矩阵 J_j

再由表 4.2 中两国的偏好信息，我们可以由系统直接分别得出两国的改良状态转移矩阵。结合冲突分析相关理论，运用第三章所构建的冲突敏感性分析系统，经计算，各状态点所对应稳定类型及均衡点如表 4.3 所示（其中“E”代表均衡）：

表 4.3 真实偏好下的稳定结果

State	Nash	GMR	SMR	SEQ
1	B	E	B	E
2	B	E	B	E
3	A	A	A	A
4	A	A	A	A
5	A	E	E	A
6	A	E	A	A
7	B	E	B	E
8		A	A	A
9		E	B	A

10		A	A	A
11		A	A	A
12		A	A	A
13	B	B	B	B

由表 4.3 可知，在真实偏好下，均衡结局集（均衡点） $E=\{s1, s2, s5, s6, s7, s9\}$ 。

4.2.2 冲突分析敏感性

近一步，我们选择均衡点 S9 对决策者 i (既 A 国) 进行 GMR 稳定性分析。

首先，我们交换决策者 A 的偏好序列中 s1 和 s2 的位置，根据 A 国的真实偏好向量，我们知道，从理论推导上状态点 S9 的稳定性应该保持不变，通过系统进行验证：

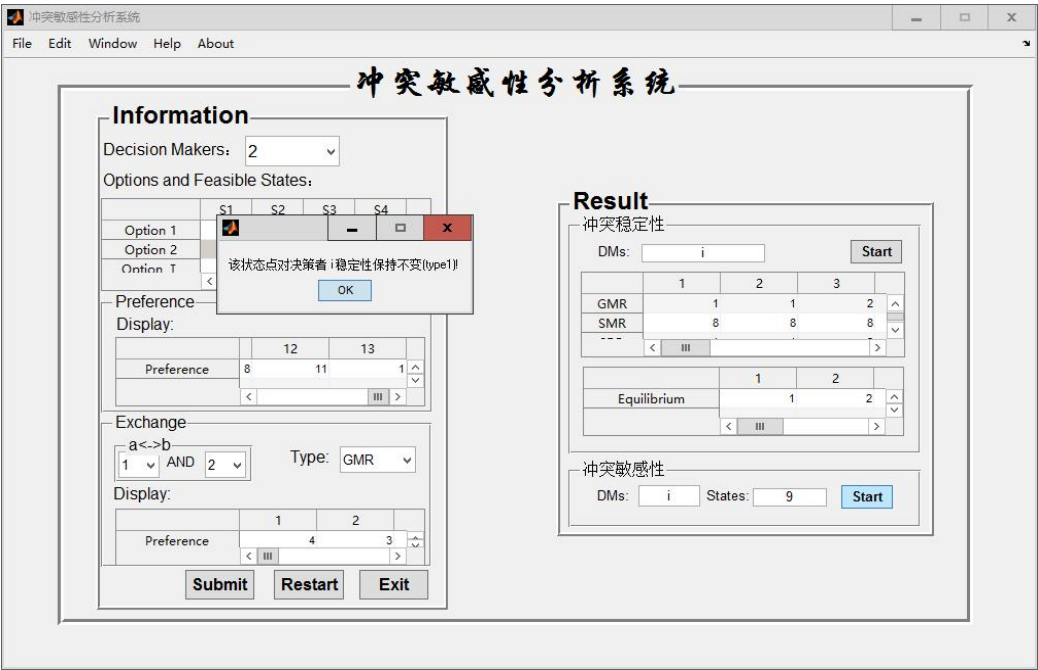


图 4.1 敏感性分析 1

系统结果与理论推导一致（其中 ‘type1’ 代表此次进行的是同侧交换）。

接下来，我们交换均衡点两侧的状态点 S5 和 S13，结果如下：

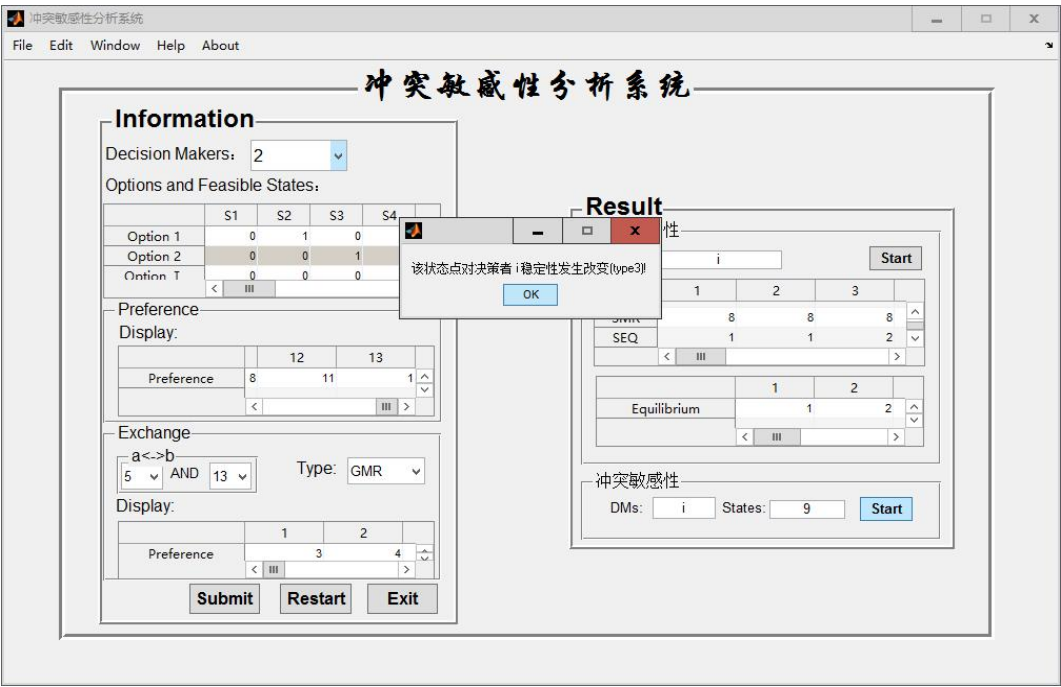


图 4.2 敏感性分析 2

计算发现，状态点 S9 的 GMR 稳定性发生变化，即从 GMR 稳定变为不稳定（其中 Type3 代表定理 7 中（2）对应的情况下）。

我们还可以得出，当交换状态点 S2 和 S11 的偏好序列时，S9 的 GMR 稳定性不发生改变（其中 type2 代表定理 7 中（1） $J_i(s, s_b)=0$ 对应的情况下）。

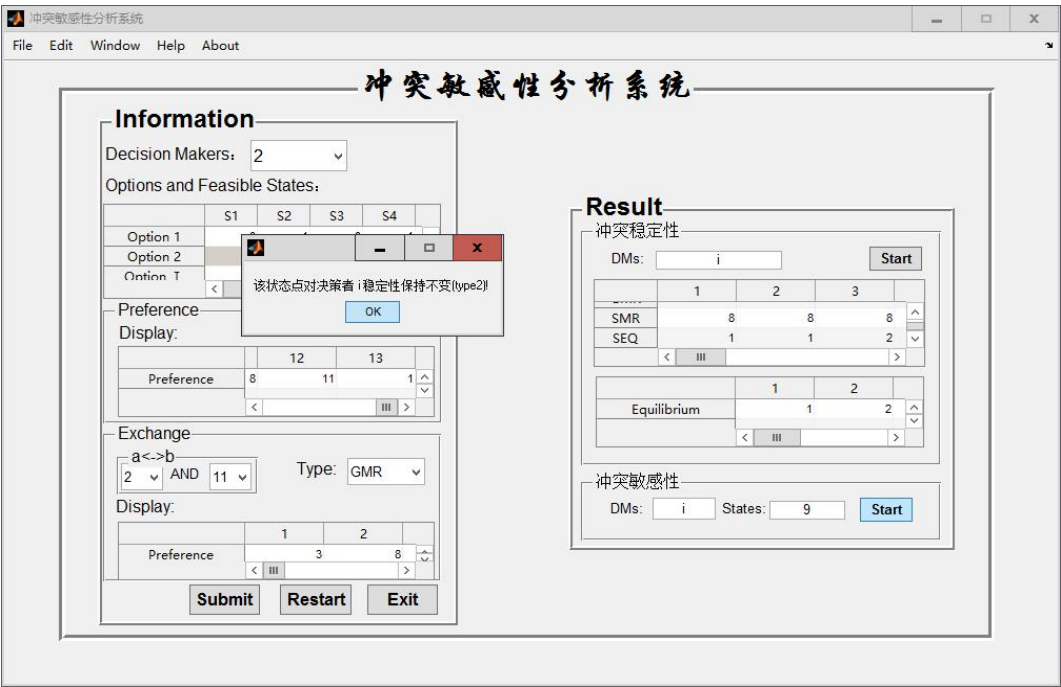


图 4.3 敏感性分析 3

至此，我们已经运用冲突敏感性分析系统进行了稳定性和敏感性分析。在稳定性分析中，我们根据方案和冲突的可能结局及双发的偏好向量得出转移矩阵，根据转移矩阵和偏好序列确定了均衡结局集；在敏感性分析中，我们选择一个均衡解，运用系统进行偏好次序的互换，观察均衡解稳定性的变化，并与理论结果进行比较。同时也说明了局中人（决策者）偏好变化可能导致稳定点的变化，系统可以在实际博弈和决策中起到辅助作用。

4.3 案例拓展

具有两个决策者的冲突博弈，就像两个人对弈，想知道对方的方案和冲突可能的结局是可行的。如果如本章前两节叙述的一样，博弈双方提前正确认知了对方的偏好，那么，博弈双方采取的方案策略和最终结局便一目了然，博弈的过程将变得毫无意义。而在现实博弈过程中，显然与我们前两节叙述的情况不符。

由于信息的不完全和错误的认知，我们对自己的偏好是完全清楚的，对对方的偏好认知难免与对手方的真实偏好向量产生差异，如表 4.4 所示。

表 4.4 A、B 两国真实以及相互认知的偏好向量

项目	符号	偏好向量(权值)
A 的真实偏好向量	V1	(3, 4, 10, 13, 6, 9, 2, 12, 5, 7, 8, 11, 1)
B 的真实偏好向量	V2	(13, 11, 5, 2, 9, 7, 12, 3, 10, 6, 4, 1, 8)
A 认知的 B 的偏好向量	V21	(13, 11, 5, 2, 8, 9, 12, 3, 10, 7, 4, 1, 6)
B 认知的 A 的偏好向量	V12	(3, 2, 10, 12, 6, 9, 1, 11, 5, 8, 7, 13, 4)

对于 A 国，进行冲突分析时，使用的偏好组合是 {V1, V21}；B 国使用的则是 {v12, v2}。进行冲突分析的结果分别见表 4.5 和表 4.6。

表 4.5 A 国认知下的稳定结果 (V1-V21)

State	Nash	GMR	SMR	SEQ
1	B	E	B	E
2	B	E	B	E
3	A	A	A	A
4	A	A	A	A
5	A	E	E	A
6	A	E	E	A
7	B	E	B	E
8		A	A	A
9		E	B	A
10	B	E	E	E
11		A	A	A

12		A	A	A
13				

$$E1=\{1, 2, 5, 6, 7, 9, 10\}$$

表 4.6 B 国认知下的稳定结果 (V2-V12)

State	Nash	GMR	SMR	SEQ
1	B	E	B	E
2	B	E	B	E
3	A	A	A	A
4		A	A	A
5	A	E	E	E
6	A	E	A	E
7	B	B	B	B
8		A	A	A
9		E	B	E
10		A	A	A
11		A	A	A
12	A	A	A	A
13	B	E	B	E

$$E2=\{1, 2, 5, 6, 9, 13\}$$

显然，E1、E2 与 E 相比都有差异，正是由于信息的不完全和错误的认知，才造成了这种差异，而这种战略误判很多时候是致命的。

通常，在实际对对手方偏好的认知中，会存在一些偏好顺序确定的状态点，同时也会存在部分次序不确定的状态点。有了敏感性分析方法，我们就能够通过对手的一种偏好次序计算出均衡点，然后根据敏感性分析方法和其他可能的偏好序列适当交换偏好次序，方便快捷的找出对大多数情况依然保持均衡的状态点，减小与真实稳定解的差距，降低决策失误的概率。

通过敏感性分析，我们还可以发现，对于一个平衡点，偏好在它前（或后）面的所有状态点无论如何重新排列组合，对其敏感性均无影响。

4.4 本章小结

本章从一个军事案例展开，进行冲突建模和系统验证，将冲突敏感性分析系统运用到实际案例中。为分析方便，案例初始情况是决策者之间信息信息是完全公开的，包括偏好信息，然而在实际冲突中，对手的偏好信息我们是很难完全预料的，是以，第三小节依据这种情况进行了必要的拓展，证明冲突敏感性分析系统在信息不完全的冲突博弈中，也是可以发挥作用的。

附录

1. MATLAB.m 文件
2. MATLAB.fig 文件
3. preference.m 文件
4. information.m 文件
5. stability.m 文件
6. sensitivei.m 文件
7. sensitivej.m 文件
8. swap.m 文件
9. Waitbar.m 文件