CASE STUDY

twenty two

IMRT

case study

OVERVIEW

CS22.1 *Application Overview and Model Development*

Radiation therapy is a common treatment for cancer. During radiation therapy, beams of radiation pass through a patient, killing both cancerous and normal cells. Therefore, the radiation therapy must be carefully planned so that a clinically prescribed dose is delivered to tumors (or targets) containing cancerous cells, while nearby organs and tissues are spared. Currently, a technique called intensity modulated radiation therapy (IMRT) is considered to be the most effective radiation therapy for many forms of cancer. In IMRT, the patient is irradiated from several beams, each of which is decomposed into hundreds of small *beamlets*, the intensities of which can be controlled individually. Typically, the physician wants to irradiate several tumor regions, and there are several nearby organs, called organs-at-risk, that must be spared. If we were to treat a patient with a single beam of radiation, it might be possible to kill all the cells in the tumor; however, it would also risk damaging normal cells in organs-at-risk located along the path of the beam. To avoid this, beams are delivered from a number of different orientations spaced around the patient so that the intersection of these beams includes the targets, which therefore receive the highest radiation dose, whereas the organs-at-risk receive radiation from some, but not all, the beams and can so be spared. (See Figure CS22.1.)

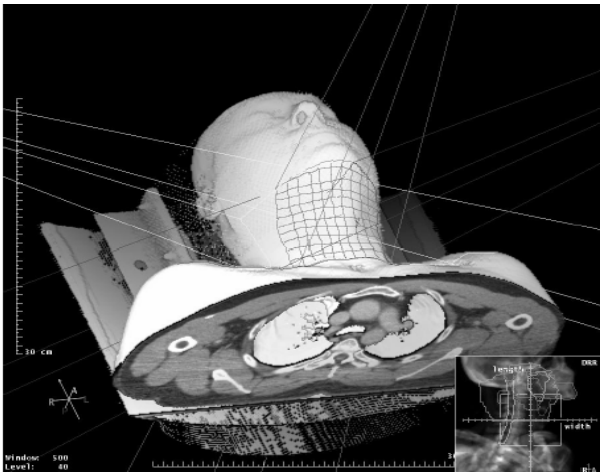

Figure CS22.1 IMRT beam intersections.

Designing an optimal radiation density profile in the patient can be approached through linear programming. The objective of this problem is to design a radiation therapy treatment plan that delivers to the targets a specific level of radiation dose, or *prescription dose* while sparing organs-at-risk by ensuring that the level of radiation does not exceed a structure-specific *tolerance dose*.

CS22.1.1 Model Definition and Assumptions

The IMRT problem is a 3-dimensional problem. In this application, however, we study a 2-dimensional approximation of it. We also only focus on the scenario in which the tumor, or target, is located in the head/neck region of the patient's body. We consider the spinal cord, saliva glands, and skin to be the nearby organs-at-risk. (See Figure CS22.2.)

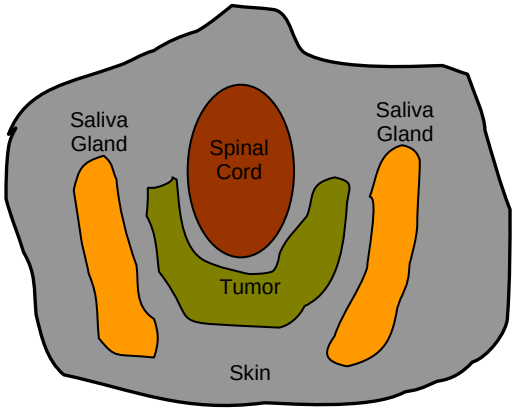

Figure CS22.2 A two-dimensional view of the tumor and the organs-at-risk in the head/neck cancer scenario.

The targets and organs-at-risk (together simply referred to as the *structures*) are irradiated with a predetermined set of *beams*, each corresponding to a particular beam angle. For dose calculation purposes, the patient body is divided into small cubicles (squares in 2-dimensions), referred to as *voxels*. (See the Excel sheet.)

j = a voxel
J_{Tumor} = set of voxels in tumor
J_{OAR} = set of voxels in OAR

In the treatment plan being studied here, the patient is irradiated from three equi-spaced beams at angles of 0^0, 120^0, and 240^0. Each beam consists of multiple beamlets, and the intensities of the beamlets can be controlled individually to obtain a desired plan. Therefore, the intensities of the beamlets are the decision variables in the linear program.

i = a beamlet
I = set of beamlets

d_{ij} = dose received by voxel j from beamlet i at unit intensity
D_j = total dose received by voxel j.

The objective function is evaluated by a pre-defined function over the dose received by the voxels. It is desirable to give a minimal dose to the organs-at-risk and to expose the tumor to the dose prescribed.

PD_j = prescription dose for any voxel $j \in J_{Tumor}$
TD_j = tolerance dose of any voxel $j \in J_{OAR}$

UB_j = upper bound on dose received by voxel j
LB_j = lower bound on dose received by voxel $j \in J_{Tumor}$

Certain penalty functions ensure that we are sending a maximum dose to the target and a minimum dose to the organs-at-risk.

P_j^O = penalty for overdosing a voxel j by unit amount

P_j^U = penalty for underdosing a voxel $j \in J_{Tumor}$ by unit amount

With these notations, the formulation can be written as a linear programming problem:

$$minimize \sum_{j \in J_{Tumor}} P_j^U w_j + \sum_{j \in J_{OAR}} P_j^O v_j + \sum_{j \in J_{OAR}} P_j^O v_j$$

subject to

$$D_j = \sum_{i \in I} d_{ij} x_i,\ j \in J_{OAR}, J_{Tumor}$$
$$D_j \le UB_j,\ j \in J_{OAR}, J_{Tumor}$$
$$D_j \ge LB_j,\ j \in J_{Tumor}$$
$$v_j \ge D_j - TD_j,\ \ j \in J_{OAR}$$
$$v_j \ge D_j - PD_j,\ j \in J_{Tumor}$$
$$w_j \ge PD_j - D_j,\ j \in J_{Tumor}$$
$$x_i \ge 0,\ i \in I,\ v_j \ge 0,\ j \in J_{OAR}, J_{Tumor},\ w_j \ge 0,\ j \in J_{Tumor}$$

We solve this problem with the Excel Solver and prepare the worksheet with decision variable cells, constraint formulas, and objective function formulas, as modeled in the formulation above.

CS22.1.2 Input

This application has two types of input: fixed input and variable input. The fixed input is:

- Number of beams = 3
- Number of beamlets = 30, and number of beamlets per beam
- Number of structures = 4
- Number of voxels = 91, and number of voxels per structure
- Dose unit intensity from each beamlet to each voxel

The variable input for the application, which is entered by the user, is:

- Overdose penalty per structure
- Underdose penalty per structure (which only applies to the tumor)
- Tolerance dose per structure (which only applies to the organs at risk)
- Prescription dose per structure (which only applies to the tumor)
- Lower bound per structure (which only applies to the tumor)
- Upper bound per structure

CS22.1.3 Output

The output for this application is the following:

- Voxel doses for each structure
- Dose-volume histogram of all structures

CS22.2 *Worksheets*

This application requires six worksheets: the welcome sheet, a model description sheet, the input sheet, a detailed input sheet, the model sheet, and the output sheet. The welcome sheet contains the title, the description of the application, and the “Start” button. (See Figure CS22.3.) The welcome sheet also contains a “View Model Description” button that takes the user to the model description sheet. (See Figure CS22.4.) The model description sheet simply describes the model to the user. The “Start” button on the

welcome sheet shows the user an options form and then navigates him or her to other sheets. (We will describe this form, referred to as the main menu form, in the next section.)

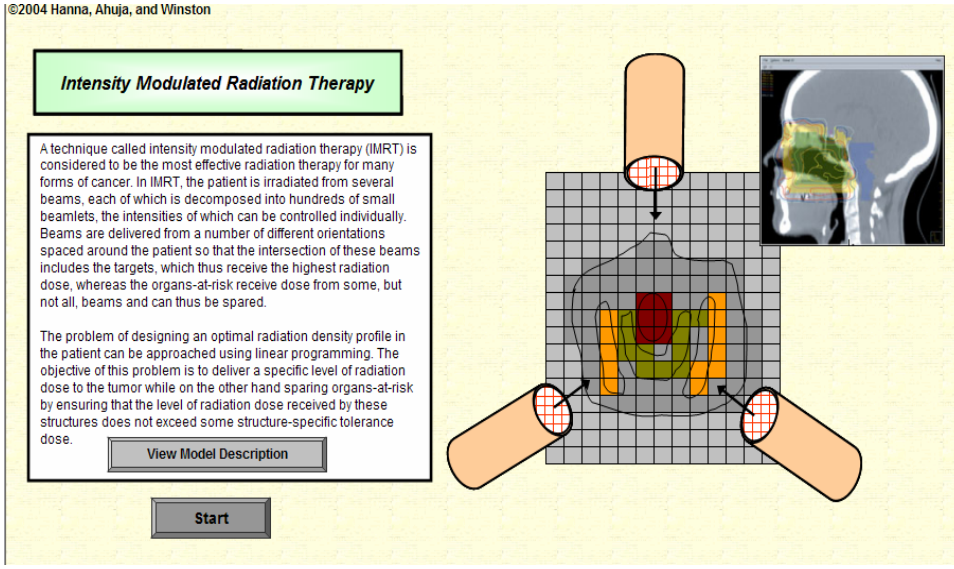

Figure CS22.3 The welcome sheet.

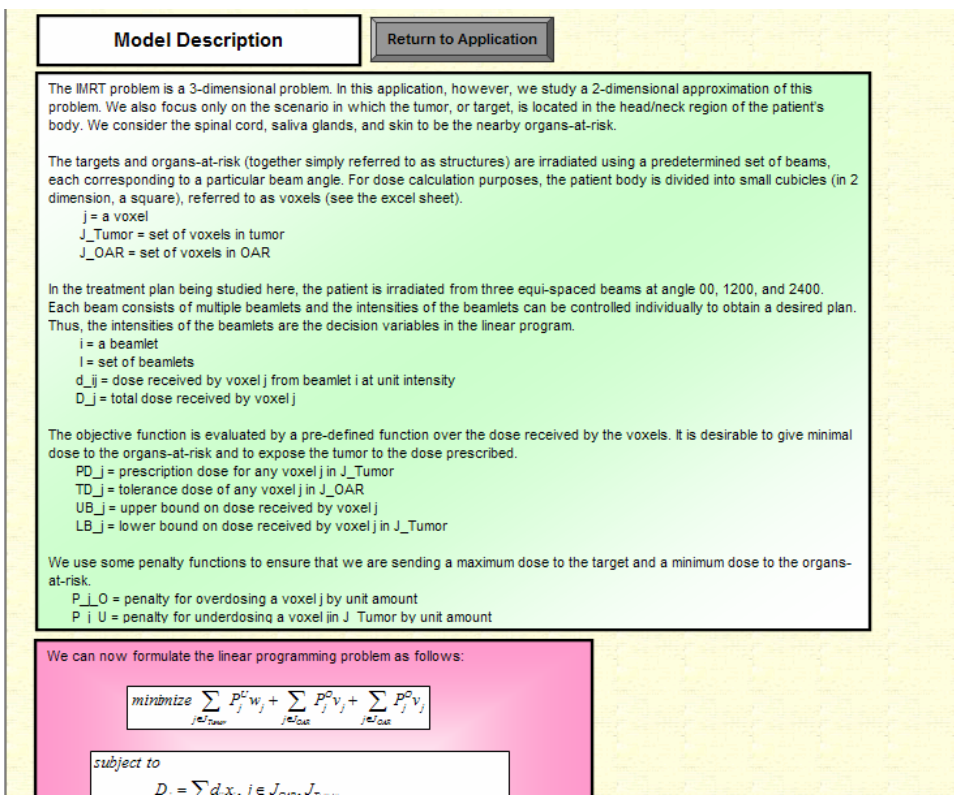

Figure CS22.4 The model description sheet.

The input sheet shows the user the fixed input and allows him or her to change the variable input. (See Figure CS22.5.) The first two tables on the sheet contain the fixed

values, and the cells in these tables are locked so the user cannot change the values. The fixed values for the dose unit intensity from each beamlet to each voxel are so large that they are stored on a separate sheet. (See Figure CS22.6.) The user can view this sheet by pressing the “View Bixel-Voxel Effect” button. The variable input values, or parameters, are shown in a table at the bottom of the input sheet. These values are also locked on the worksheet so the user modifies them with a form. The user can view this form by pressing the “Modify Values” button. We ensure that the user provides the input with a form so that bounds are placed on the possible input values. (We will discuss these input forms in the next section.)

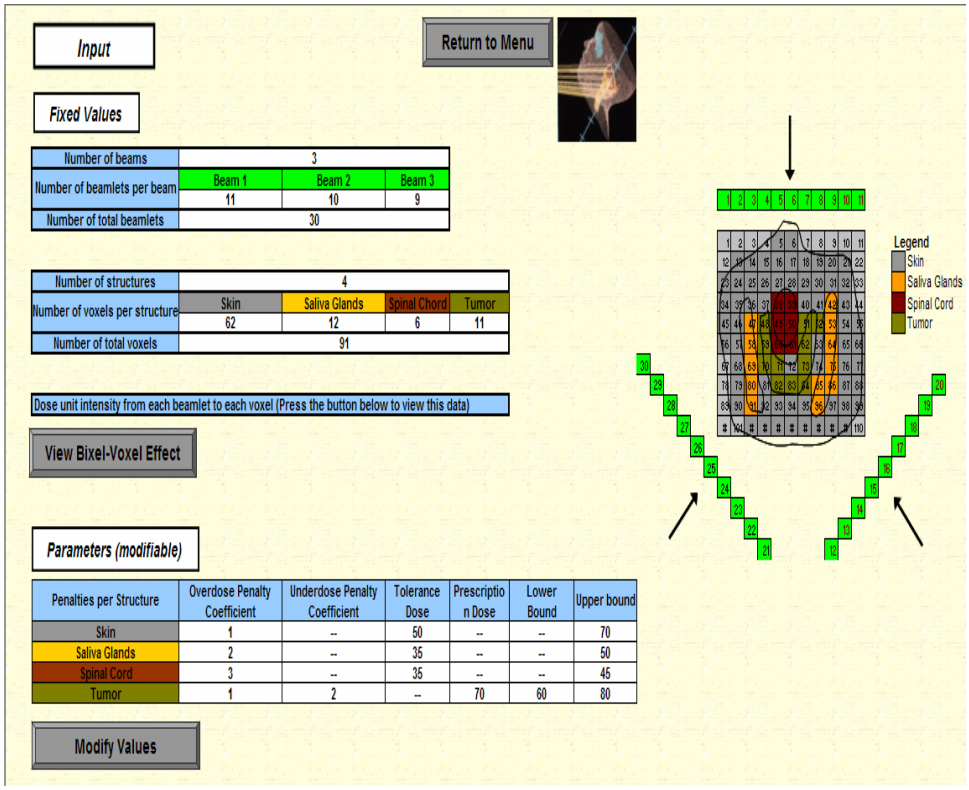

Figure CS22.5 The input sheet.

The model sheet is the most complex sheet of the application as it contains all of the components of the IMRT linear programming model. (See Figure CS22.7.) At the top of the page, there are two sets of decision variable cells: bixel intensities and voxel doses. The voxel dose decision variables are actually calculated variables; in other words, they are defined in terms of the bixel intensity variables. The bixel-voxel fixed input values and the SUMPRODUCT function create this definition.

The constraint formulas are contained in several cells on the worksheet. With the variable input provided by the user, the application calculates the overdose penalties, the tolerance/prescription dose penalties, the upper bound and lower bound penalties, and the dose deviations for each structure. The objective function cell sums the total penalties per structure, a value that we seek to minimize. Even though this model seems large and slightly complex in its constraints, it is still a linear programming problem and can easily be solved by the Excel Solver.

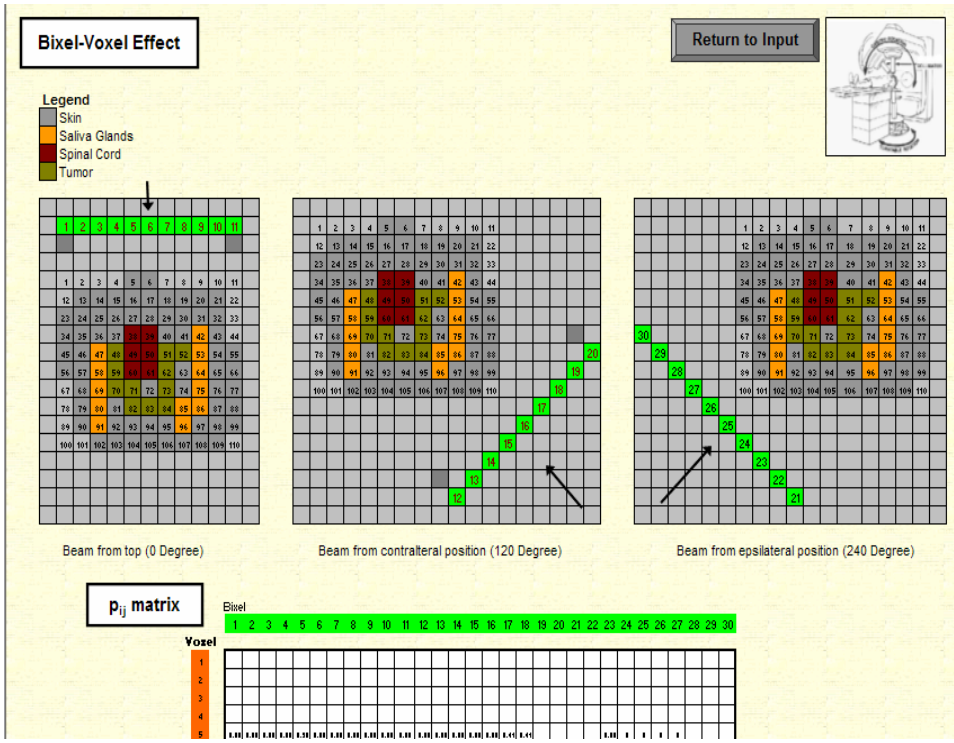

Figure CS22.6 The Bixel-Voxel data sheet.

The output sheet shows the voxel dose per structure as well as a dose-volume histogram. (See Figure CS22.8.) The voxel doses are displayed in a grid that maps the voxels to their structures, and a legend is provided for the structures. The dose-volume histogram presents the cumulative percentage of the dose per volume of each structure; here, the volume is measured by number of voxels. For example, in Figure CS22.8, 36.6 percent of the tumor voxels are receiving a dose of 70 or higher.

Summary

Welcome sheet	Contains the application description and the “View Model Description” and “Start” buttons.
Model description sheet	Contains a detailed description of the IMRT linear programming model.
Input sheet	Contains the fixed and variable input values and the “View Bixel-Voxel Effect” and “Modify Values” buttons.
Detailed input sheet	Contains the detailed input for the bixel-voxel unit intensities.
Model sheet	Contains the decision variable cells, the constraint formulas, and the objective function formula for the IMRT linear programming model.
Output sheet	Contains the grid of the voxel dosage for all the structures and the dose-volume histogram.

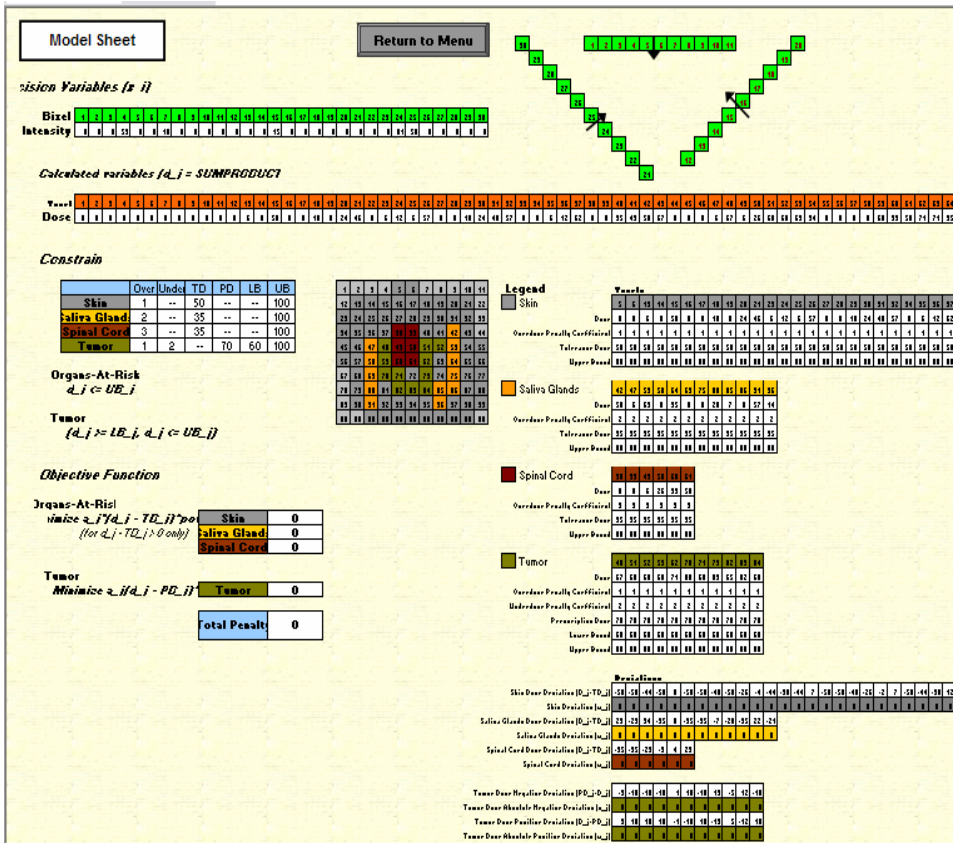

Figure CS22.7 The model sheet.

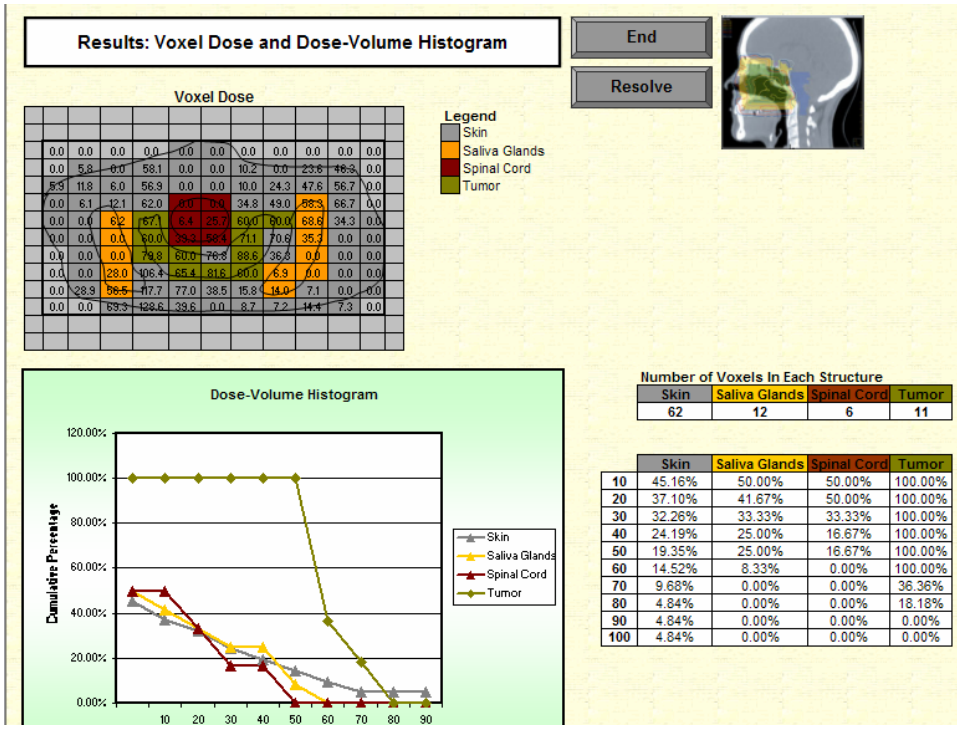

Figure CS22.8 The results sheet.

CS22.3 *User Interface*

For this application's user interface, we use navigational buttons and several user forms. The first user form, the main menu form, allows the user to navigate to the input sheet to view or modify input values, the model sheet to visualize the model, and the output sheet to solve the problem and see the results. (See Figure CS22.9.) This form requires a frame with three option buttons.

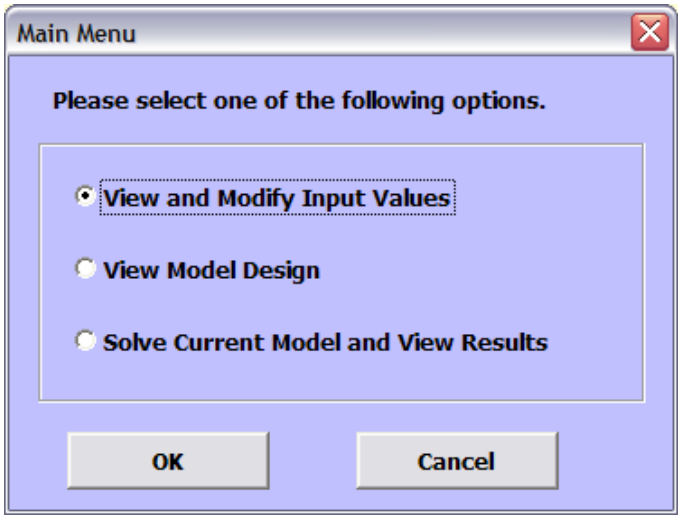

Figure CS22.9 The main menu user form.

Several forms allow the user to enter or modify the input values. The first form is the main input form shown in Figure CS22.10. In this form, the user chooses whether to modify the input for the organs-at-risk (OAR) or the tumor. The main input form requries a frame and two option buttons.

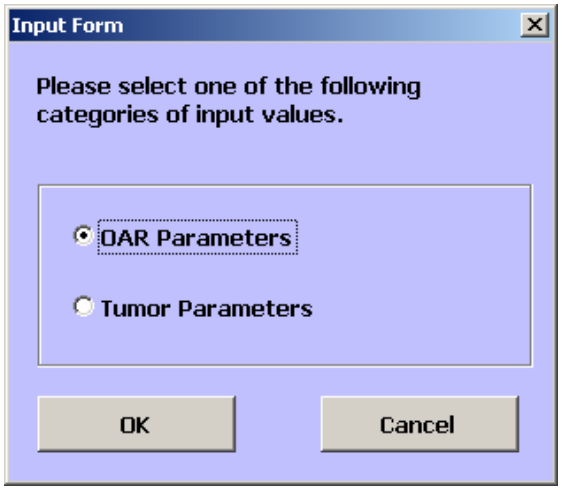

Figure CS22.10 The input form.

The OAR structure input and the tumor input each have a separate form. Figure CS22.11 presents the form for the OAR structures. This form uses a tab structure to allow the user to change the penalty coefficient, the tolerance dose, and the upper bound for any of the OAR structures. The values and bounds for the scroll bar and text box controls on the tabs vary per structure. [See Figures CS25.11(a), (b), and (c).]

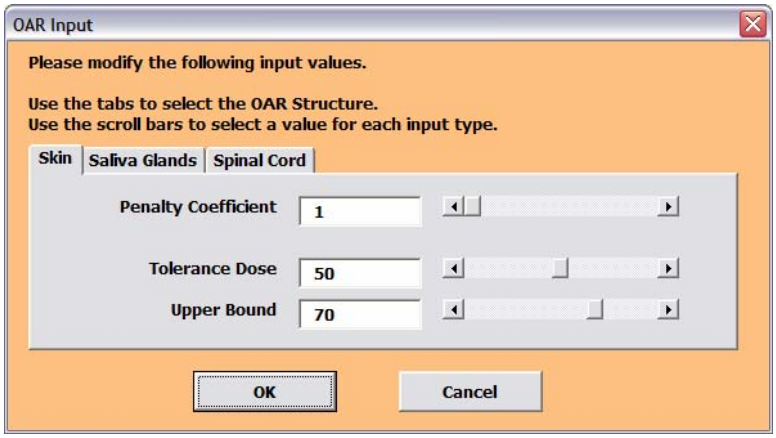

Figure CS22.11 (a)

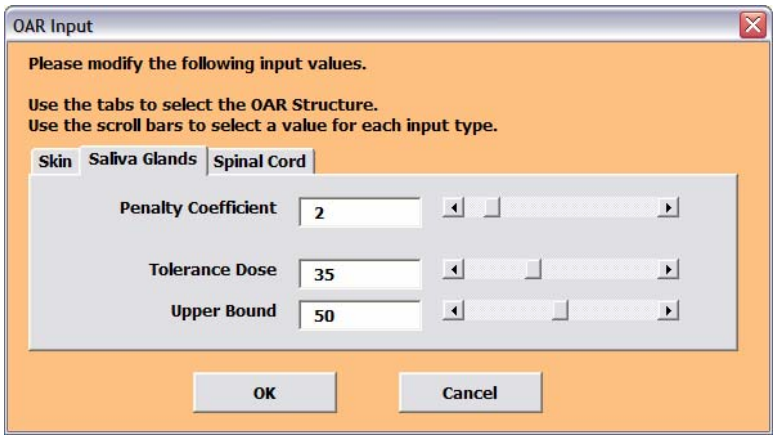

Figure CS22.11 (b)

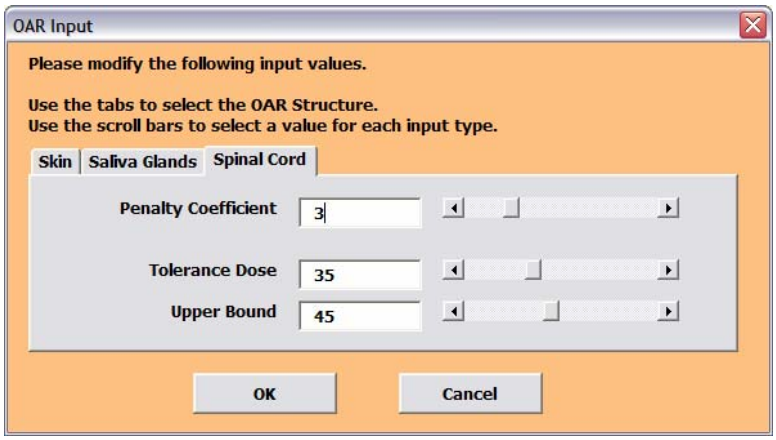

Figure CS22.11 (c)

Figure CS22.11 The OAR input form.

The input form for the tumor uses one set of scroll bars and text boxes to receive the values for the overdose and underdose penalty coefficients, the prescription dose, and the lower and upper bounds of the tumor dosage. (See Figure CS22.12).

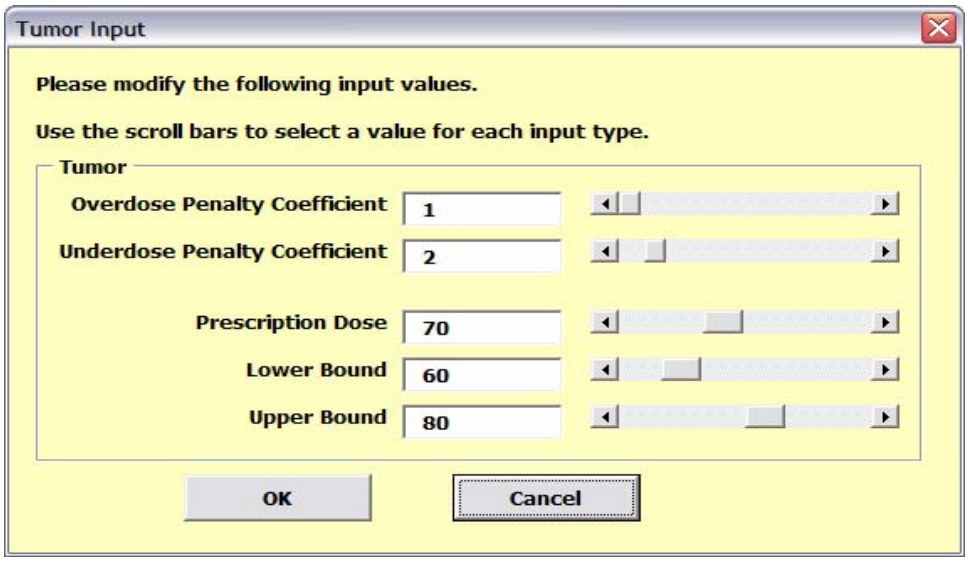

Figure CS22.12 The tumor input form.

Summary	**Main menu form**	Where the user can navigate to the input sheet to view or modify input values, the model sheet to visualize the model, and the output sheet to solve the problem and see the results.
	Main input form	Where the user can choose to enter/modify the input for the OAR structures or tumor.
	OAR input form	Where the user can use tabs to enter/modify the input for any of the OAR structures.
	Tumor input form	Where the user enters/modifies the input for the tumor.
	Navigational buttons	"Start," "View Model Description," "View Bixel-Voxel Effect," "Return to Menu," "Return to Input," "End."
	Functional buttons	"Modify Values," "Re-solve."

CS22.4 *Procedures*

We will now outline the procedures for this application beginning with the initial sub procedures. (See Figure CS22.13.) The *Main* procedure, which is called from the "Start" button, shows the main menu form. The "Modify Values" button on the input sheet calls the *ModifyInput* procedure, which displays the main input form.

```
Option Explicit

Sub MAIN()  'called from Start button and Return to Menu buttons
    Worksheets("Welcome").Visible = True
    Worksheets("Input").Visible = False
    Worksheets("Model").Visible = False

    frmMainMenu.Show
End Sub

Sub ModifyInput()   'called from Modify Input button on input sheet
    frmInput.Show
End Sub
```

Figure CS22.13 The *Main* procedure and the *ModifyInput* procedure.

The procedures for the main menu form simply navigate the user to the sheet that corresponds to the selected option button choice. (See Figure CS22.14.) If the user chooses to solve the model and see the results, then the *SolveModel* procedure is also called.

```
'main menu navigations

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    If optViewModifyInput Then
        Worksheets("Input").Visible = True
        Worksheets("Welcome").Visible = False
    ElseIf optViewModel Then
        Worksheets("Model").Visible = True
        Worksheets("Welcome").Visible = False
    ElseIf optSolveResults Then
        Call SolveModel
        Worksheets("DVH").Visible = True
        Worksheets("Welcome").Visible = False
    End If

    Unload Me
End Sub
```

Figure CS22.14 The procedures for the main menu form.

The procedures for the main input form simply show the user the corresponding input form for his or her option button choice. (See Figure CS22.15.)

```
'main input form

Private Sub cmdCancel_Click()
    Unload Me
End Sub

Private Sub cmdOK_Click()
    If optOAR Then
        Unload Me
        frmOAR.Show
    ElseIf optTumor Then
        Unload Me
        frmTumor.Show
    End If

End Sub
```

Figure CS22.15 The procedures for the main input form.

There are several procedures for the OAR input form. (See Figures CS25.16, CS25.17, and CS25.18.) Figure CS22.16 presents the procedures that record the input values after the user has clicked the “OK” button on the form. Figure CS22.17 presents the procedures that coordinate the scroll bars and text boxes for all of the input values on all of the tabs. Figure CS22.18 presents the initialization procedure that sets the default values for the text boxes and scroll bars on all the tabs.

```
'OAR input

Private Sub cmdCancel_Click()
    Unload Me
    frmInput.Show
End Sub

Private Sub cmdOK_Click()
    'check bound values
    If txtSkinUB.Value < txtSkinTD.Value Or _
            txtSalUB.Value < txtSalTD.Value Or _
            txtSpinUB.Value < txtSpinTD.Value Then
        MsgBox "All Upper Bounds must be greater than or equal to the Tolerance Doses."
        Exit Sub
    End If

    'copy new values to Input sheet
    With Range("SkinStart")
        .Offset(0, 1).Value = scrSkinPen.Value
        '.Offset(0, 2).Value = scrSkinPow.Value
        .Offset(0, 3).Value = scrSkinTD.Value
        .Offset(0, 6).Value = scrSkinUB.Value
    End With

    With Range("SalStart")
        .Offset(0, 1).Value = scrSalPen.Value
        '.Offset(0, 2).Value = scrSalPow.Value
        .Offset(0, 3).Value = scrSalTD.Value
        .Offset(0, 6).Value = scrSalUB.Value
    End With

    With Range("SpinStart")
        .Offset(0, 1).Value = scrSpinPen.Value
        '.Offset(0, 2).Value = scrSpinPow.Value
        .Offset(0, 3).Value = scrSpinTD.Value
        .Offset(0, 6).Value = scrSpinUB.Value
    End With

    Unload Me
End Sub
```

Figure CS22.16 The first part of the procedures for the OAR input form.

Figures CS25.19 and CS25.20 present the procedures for the tumor input form. The procedures in Figure CS22.19 record the final tumor values once the user has clicked the “OK” button. The procedures in Figure CS22.20 coordinate the scroll bars and text boxes and initialize the form values.

```
Private Sub scrSalPen_Change()
    txtSalPen.Value = 2 ^ scrSalPen.Value
End Sub

Private Sub scrSalTD_Change()
    txtSalTD.Value = scrSalTD.Value
End Sub

Private Sub scrSalUB_Change()
    txtSalUB.Value = scrSalUB.Value
End Sub

Private Sub scrSkinPen_Change()
    txtSkinPen.Value = 2 ^ scrSkinPen.Value
End Sub

Private Sub scrSkinTD_Change()
    txtSkinTD.Value = scrSkinTD.Value
End Sub

Private Sub scrSkinUB_Change()
    txtSkinUB.Value = scrSkinUB.Value
End Sub

Private Sub scrSpinPen_Change()
    txtSpinPen.Value = 2 ^ scrSpinPen.Value
End Sub

Private Sub scrSpinTD_Change()
    txtSpinTD.Value = scrSpinTD.Value
End Sub

Private Sub scrSpinUB_Change()
    txtSpinUB.Value = scrSpinUB.Value
End Sub
```

Figure CS22.17 The second part of the procedures for the OAR input form.

```
Private Sub UserForm_Initialize()
    'initialize form with current input values
    With Range("SkinStart")
        scrSkinPen.Value = .Offset(0, 1).Value
        'scrSkinPow.Value = .Offset(0, 2).Value
        scrSkinTD.Value = .Offset(0, 3).Value
        scrSkinUB.Value = .Offset(0, 6).Value
    End With

    With Range("SalStart")
        scrSalPen.Value = .Offset(0, 1).Value
        'scrSalPow.Value = .Offset(0, 2).Value
        scrSalTD.Value = .Offset(0, 3).Value
        scrSalUB.Value = .Offset(0, 6).Value
    End With

    With Range("SpinStart")
        scrSpinPen.Value = .Offset(0, 1).Value
        'scrSpinPow.Value = .Offset(0, 2).Value
        scrSpinTD.Value = .Offset(0, 3).Value
        scrSpinUB.Value = .Offset(0, 6).Value
    End With

    txtSkinPen.Value = scrSkinPen.Value
    'txtSkinPow.Value = scrSkinPow.Value
    txtSkinTD.Value = scrSkinTD.Value
    txtSkinUB.Value = scrSkinUB.Value

    txtSalPen.Value = scrSalPen.Value
    'txtSalPow.Value = scrSalPow.Value
    txtSalTD.Value = scrSalTD.Value
    txtSalUB.Value = scrSalUB.Value

    txtSpinPen.Value = scrSpinPen.Value
    'txtSpinPow.Value = scrSpinPow.Value
    txtSpinTD.Value = scrSpinTD.Value
    txtSpinUB.Value = scrSpinUB.Value
End Sub
```

Figure CS22.18 The third part of the procedures for the OAR input form.

```
'tumor input form

Private Sub cmdCancel_Click()
    Unload Me
    frmInput.Show
End Sub

Private Sub cmdOK_Click()
    'check bound values
    If txtLB.Value > txtPD.Value Or txtUB.Value < txtPD.Value Then
        MsgBox "The Lower Bound must be less than the Prescription Dose and " & _
            "the Upper Bound must be greater than the Prescription Dose."
        Exit Sub
    End If

    'record values to Input sheet
    With Range("TumStart")
        .Offset(0, 1).Value = scrPenCoeffO.Value
        '.Offset(0, 2).Value = scrPenPow.Value
        .Offset(0, 4).Value = scrPD.Value
        .Offset(0, 5).Value = scrLB.Value
        .Offset(0, 6).Value = scrUB.Value
    End With

    Unload Me
End Sub
```

Figure CS22.19 The first part of the procedures for the tumor input form.

```
Private Sub scrLB_Change()
    txtLB.Value = scrLB.Value
End Sub

Private Sub scrPD_Change()
    txtPD.Value = scrPD.Value
End Sub

Private Sub scrPenCoeffO_Change()
    txtPenCoeffOver.Value = 2 ^ scrPenCoeffO.Value
End Sub

Private Sub scrPenCoeffU_Change()
    txtPenCoeffUnder.Value = 2 ^ scrPenCoeffU.Value
End Sub

Private Sub scrUB_Change()
    txtUB.Value = scrUB.Value
End Sub

Private Sub UserForm_Initialize()
    With Range("TumStart")
        scrPenCoeffO.Value = .Offset(0, 1).Value
        scrPenCoeffU.Value = .Offset(0, 2).Value
        'scrPenPow.Value = .Offset(0, 2).Value
        scrPD.Value = .Offset(0, 4).Value
        scrLB.Value = .Offset(0, 5).Value
        scrUB.Value = .Offset(0, 6).Value
    End With

    txtPenCoeffOver.Value = scrPenCoeffO.Value
    txtPenCoeffUnder.Value = scrPenCoeffU.Value
    'txtPenPow.Value = scrPenPow.Value
    txtPD.Value = scrPD.Value
    txtLB.Value = scrLB.Value
    txtUB.Value = scrUB.Value

End Sub
```

Figure CS22.20 The second part of the procedures for the tumor input form.

The *SolveModel* procedure solves the model prepared in the model sheet. (See Figure CS22.21.) The decision variables are entered as a large non-adjacent range; note how the names of the ranges are provided. The constraints have been grouped together based on their inequalities so there are only three *SolverAdd* commands. The result of the Solver solution is stored in case the input yields an infeasible result.

```
Sub SolveModel()
    Dim result As Integer

    Application.ScreenUpdating = True
    Worksheets("Model").Visible = True
    Worksheets("Model").Activate

    '**note that constraints and objective function are in terms of voxels
    'voxels are calculated variables from bixel dosage (DV) and pij matrix

    SolverReset
    'objective function = total penalty, decision variables = beam bixel dosage
    SolverOK SetCell:=Range("ObjFunc"), MaxMinVal:=2, _
    ByChange:=Range("DV_Bixels,SkinViolation,GlandViolation,CordViolation, TumorPositiveDeviation, TumorNegativeDeviation")

    'OARs: voxel dose less than or equal to UB
    SolverAdd CellRef:=Range("AllVoxels"), Relation:=1, FormulaText:=Range("AllUB")
        'SolverAdd CellRef:=Range("SkinVoxels"), Relation:=1, FormulaText:=Range("SkinUB")
        'SolverAdd CellRef:=Range("SalVoxels"), Relation:=1, FormulaText:=Range("SalUB")
        'SolverAdd CellRef:=Range("SpinVoxels"), Relation:=1, FormulaText:=Range("SpinUB")

    'Tumor: voxel dose less than or equal to UB and greater than or equal to LB
    SolverAdd CellRef:=Range("TumorVoxels"), Relation:=3, FormulaText:=Range("TumorLB")
        'SolverAdd CellRef:=Range("TumorVoxels"), Relation:=1, FormulaText:=Range("TumorUB")

    'violations should be >= dose deviations
    SolverAdd CellRef:=Range("AllViolations"), Relation:=3, FormulaText:=Range("AllDoseDeviations")

    'decision variables should be greater than or equal to 0
    SolverOptions AssumeNonNeg:=True

    result = SolverSolve(UserFinish:=True)
    If result = 5 Then
        MsgBox "No feasible solution was found. Please modify your input values and RESOLVE."
    End If

    Worksheets("Model").Visible = False
    Range("ResultsVoxels").Columns.AutoFit
    Range("A1").Select
    Application.ScreenUpdating = True
End Sub
```

Figure CS22.21 The *SolveModel* procedure.

Figure CS22.22 presents the navigational procedures for the application's buttons.

```
Sub EndButton()
    Worksheets("Welcome").Visible = True
    ActiveSheet.Visible = False
End Sub

Sub ViewBixelVoxel()
    Worksheets("Bixel-Voxel Effect").Visible = True
    Worksheets("Input").Visible = False
End Sub

Sub ReturnToInput()
    Worksheets("Input").Visible = True
    Worksheets("Bixel-Voxel Effect").Visible = False
End Sub

Sub ViewDescrip()
    Worksheets("Descrip").Visible = True
    Worksheets("Welcome").Visible = False
End Sub

Sub ReturnApp()
    Worksheets("Welcome").Visible = True
    Worksheets("Descrip").Visible = False
End Sub
```

Figure CS22.22 The navigational procedures.

Summary		
	Main	Initializes the application and shows the user the main menu form.
	ModifyInput	Shows the user the main input form.
	Procedures for the main menu form	Navigate the user to various sheets; also call the *SolveModel* procedure when the user goes to the output sheet.
	Procedures for the main input form	Show the user the OAR or the tumor input form.
	Procedures for the OAR input form	Record the input for the OAR structures.
	Procedures for the tumor input form	Record the input for the tumor.
	SolveModel	Runs Solver to solve the IMRT model.
	Navigational procedures	For "View Model Description," "View Bixel-Voxel Effect," "Return to Menu," "Return to Input," and "End."

CS22.5 *Re-solve Options*

The user can re-solve this application by pressing the "Re-solve" button on the output sheet. This button is assigned to the *Re-solve* procedure, which displays the main input form and then calls the *SolveModel* procedure. (See Figure CS22.23.) The user can then modify input values for any OAR structure or for the tumor and view the updated results.

```
Sub Resolve()    'called from Resolve button on results sheet
    frmInput.Show
    Call SolveModel
End Sub
```

Figure CS22.23 The *Re-solve* procedure.

Summary		
	"Re-solve" button	Re-displays the input forms to the user and re-solves the model.

CS22.6 *Summary*

- This application seeks to solve the IMRT problem by designing a radiation therapy treatment plan that delivers a specific level of radiation dose, or *prescription dose*, to the targets, while sparing organs-at-risk by ensuring that the level of radiation they receive does not exceed a structure-specific *tolerance dose*.
- This application requires six worksheets: the welcome sheet, a model description sheet, the input sheet, a detailed input sheet, the model sheet, and the output sheet.
- For this application's user interface, we use navigational buttons and several user forms.
- Several application procedures collect the input for the model and solve the model with the Solver commands.

- The user can re-solve the application by pressing the “Re-solve” button on the output sheet. The user can then modify the input values for any OAR structure or for the tumor and view the updated results.

CS22.7 *Extensions*

- Suggest an alternative way for the user to enter the variable input. Consider designing one of the sheets to have controls on the worksheet.
- Allow the user to change some of the fixed input values.
- Create a grid on the output sheet that reflects the “hot” areas from the radiation. In other words, create a color scheme that reflects high dosage areas on a voxel grid for all the structures.