

Contents

1 What is Software Visualization	1
1.1 Introduction.....	1
1.2 Basics of Graphical User Interfaces.....	2
1.3 Software Visualization	4
1.3.1 Classification of Software Visualization	8
1.4 Algorithm Animation	10
1.4.1 Event-Driven Approach	11
1.4.2 Data-Driven Approach	11
1.5 Advanced Graphics Algorithms	12
2 The Stable Matching Problems	14
2.1 Introduction.....	14
2.2 Basic Definitions for Stable Matching	14
2.3 Types of Stable Matching.....	19
2.3.1 Non- Bipartite Matching	20
2.3.2 Bipartite Matching.....	20
2.4 Stable Matching applications in real-life.....	21
2.4.1 House Allocation Problem.....	22
2.4.2 Capacitated House Allocation Problem (CHA)	24
2.4.3 Stable Marriage Problems.....	25
2.4.4 Hospitals-Residents Problem (HR).....	31
2.4.5 Student Project Allocation Problem.....	35
2.4.6 Stable Roommates Problem (SR)	35
3 Methods of Solving the Student Project Allocation Problem.....	38
3.1 Introduction.....	38
3.2 Why Student Project Allocation?	38
3.3 Simplified Model for Student-Project Allocation Problem	39
3.4 One Side Student Project Allocation Problem.....	44

3.5 Two Sided Student Project Allocation Problem	46
3.5.1 Student-Project Allocation Problem with preference lists over students (SPA-Model)	47
3.5.2 Student-Project Allocation Problem with preference lists over Projects (SPA-P).....	54
 4 Student Project Allocation with Preference Lists Over (Student, Project) Pairs.....	61
4.1 Introduction.....	61
4.2 The SPA-(s, p) model.....	62
4.2.1 SPA-(s, p) model definition	62
4.2.2 Student-oriented algorithm for SPA-(s, p) model.....	68
4.3 Data Structure for SPA-(s, p) Model	76
4.4 SPA-(s, p) is a more general model	80
4.4.1 SPA model reduces to SPA-(s, p) model	81
4.4.2 SPA-P model reduces to SPA-(s, p) model.....	87
 5 Visualization for Student Project Allocation with Preferences over Pairs.....	91
5.1 Introduction.....	91
5.2 The SPA-(s, p) model.....	92
5.3 SPA-(s, p)-Student (<i>I</i>) visualization	93
5.3.1 The Creation of the Instances	95
5.3.2 The Algorithm Tracing.....	100
 Conclusion	112
Reference	115

List of Figures

1.1 An example for GUI	2
1.2 Call graph	5
1.3 Control flow graph	6
1.4 X-Tango animation of the quicksort algorithm	7
1.5 Taxonomy introduced by Myers	8
1.6 linear insertion sorting and heap sorting in the videotape “Sorting out Sorting”	9
1.7 Three-dimensional animation of bubble sort	13
2.1 graph with matching and augmenting path	15
2.2 complete preference list and incomplete preference lists	17
2.3 An example on stable matching problem with complete lists	18
2.4 An instance of HA	23
2.5 An instance of CHA	24
2.6 Instance 1 of SM	26
2.7 Instance 2 of SM	28
2.8 Instance 3 of SM	28
2.9 An instance 4 of HR	33
2.10 An instance of SR	36
3.1 An instance I_1 of the simplified student-project allocation problem	40
3.2 Network flow for instance I_1	43
3.3 An instance I_2 of the one side matching for student-project allocation problem	45
3.4 Network flows for instance I_2	46
3.5 An instance I_3 of the student-project allocation problem	48
3.6 An instance I_4 of the student-project allocation problem	55
4.1 An instance I_1 of SPA-(s, p)	62
4.2 Preference lists to create an instance I_2 of SPA-(s, p)	63
4.3 MAIN SPA-(s, p)-student algorithm	71
4.4 An instance of SPA-(s, p) model	77
4.5 The lecturers’ preference lists in the main array and with first and last pointers	78
4.6 The projects’ preference lists in the main array and with first and last pointers	79
4.7 The students’ preference lists in the main array and with first and last pointers	79
4.8 The new instance $f_1(I)$	81
4.9 The new instance $f_2(I)$	88
5.1 An instance of SPA-(s, p) model	93

5.2 The program start window, part “a” is the visual panel and part “b” is the utility panel	94
5.3 The program utility panel	95
5.4 The program window after clicking “File Problem” button	96
5.5 The program window after clicking “File Problem” button	97
5.6 The visualization of student preference list	98
5.7 The visualization of lecturer l_1 preference list and his projects’ preference lists	98
5.8 The visualization of the complete instance	99
5.9 Student s_1 chooses the first project in his preference	101
5.10 Student s_1 applied to project p_1 , these lead to provisional assignments between them	102
5.11 The lecturer l_1 and project p_1 become full and their corresponding names are colored orange	104
5.12 The preference lists after deleting the first pair (s_5, p_2)	105
5.13: The preference lists after deleting the two pairs (s_5, p_2) and (s_3, p_2)	106
5.14 lecturer l_1 and project p_1 are over-subscribed, the worst student s_2 that was assigned to the project p_1 is selected and the edge between them is flashing	107
5.15 preference lists after break assignment between the student s_2 and the project p_1	109
5.16 The stable matching	110