A genetic Approach to the N-Queen Problem

Mohammad Bahrami - 9724133
Masoud Mousavi - 9735453
AmirHossein Ahmadi - 9722343

July 16, 2021

---

1 Algorithm

1.1 Initialization

Each individual is a vector of size N filled with a random permutation of numbers between 1 and N. Each element shows the column of the queen for the element's index row.

1.2 Evaluation

We count the number of distinct queens guarding each other in a diagonal matter. There is no need to check for the row and the column because we never let the vector to have any repeated element. We use two methods for evaluation, Roulette wheel and Tournament evaluation. Then, we divide the guards by the Maximum possible guards and subtract the result from 1 to create a maximization problem.

$$MaxGuards = \frac{N(N-1)}{2}$$

Roulette wheel: We calculate the sum of the fitness of the generation and create a wheel with it. Then, we pick a random number between 0 and the sum of fitnesses. we subtract the fitness of each individual from the number and when it reaches 0 or below, we pick the last element that we encountered.

Tournament: We pick two random individuals from the population and choose the one with a better fitness.

1.3 Crossover

Partially Mapped Crossover: First, a Start and an End point is randomly chosen. Then, for each element from start to end, the element is replaced by the content of the other parent. Then, the element that has been removed will replace the element that has come from the other parent.

Order One Crossover: First, we choose a random start and end point. We Copy everything from each parent to relevant child between the start and ending point. Then, we traverse the other parent and put any number that is currently not in the child by order.

1.4 Mutation

Swap Mutation: We randomly select two elements and swap them.

Displacement Mutation: We randomly select a point and an element. Then we move that element to the selected point and shift the other elements relatively.

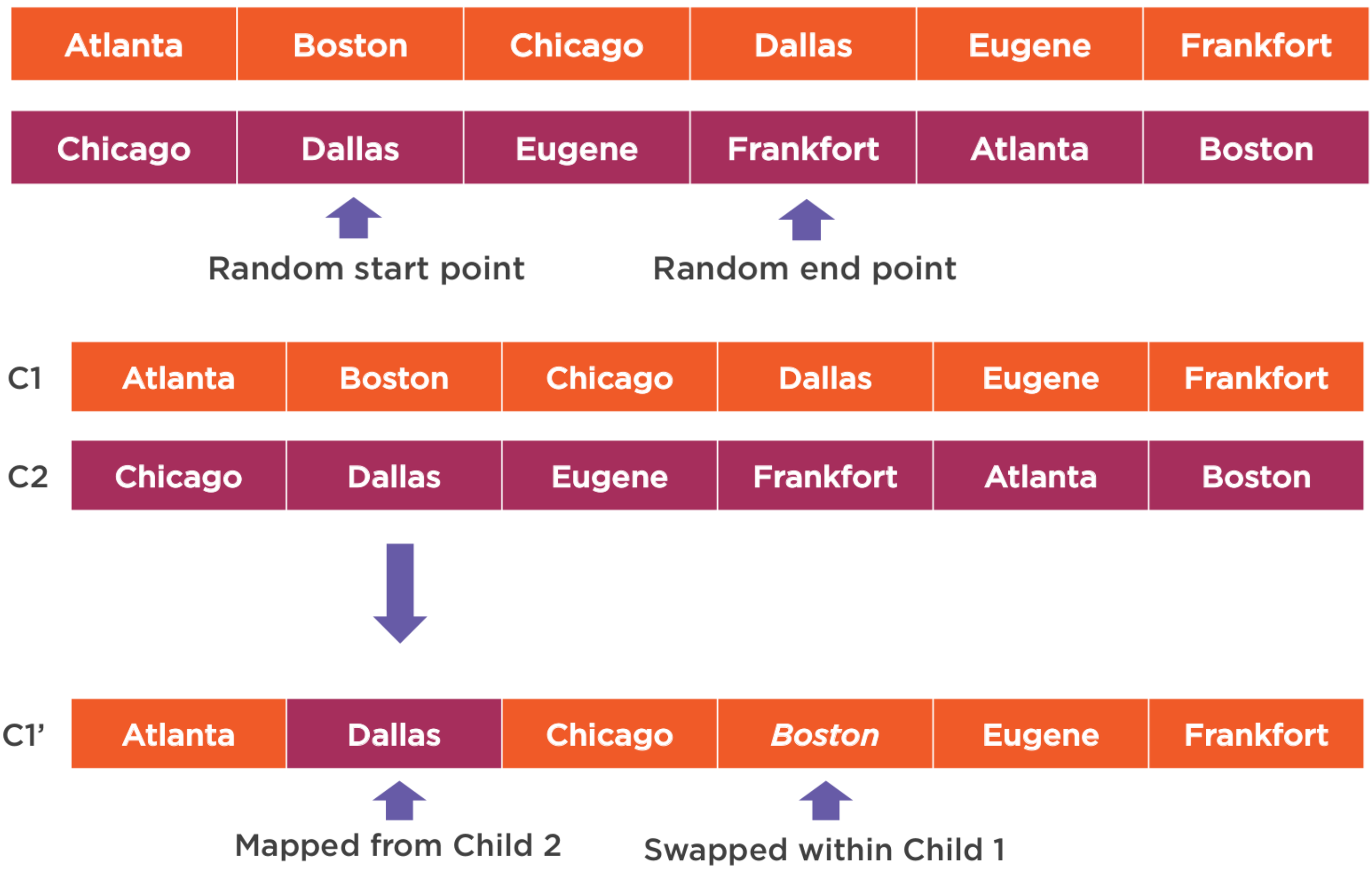

Figure 1: Demonstration of one step of Partially Mapped Crossover

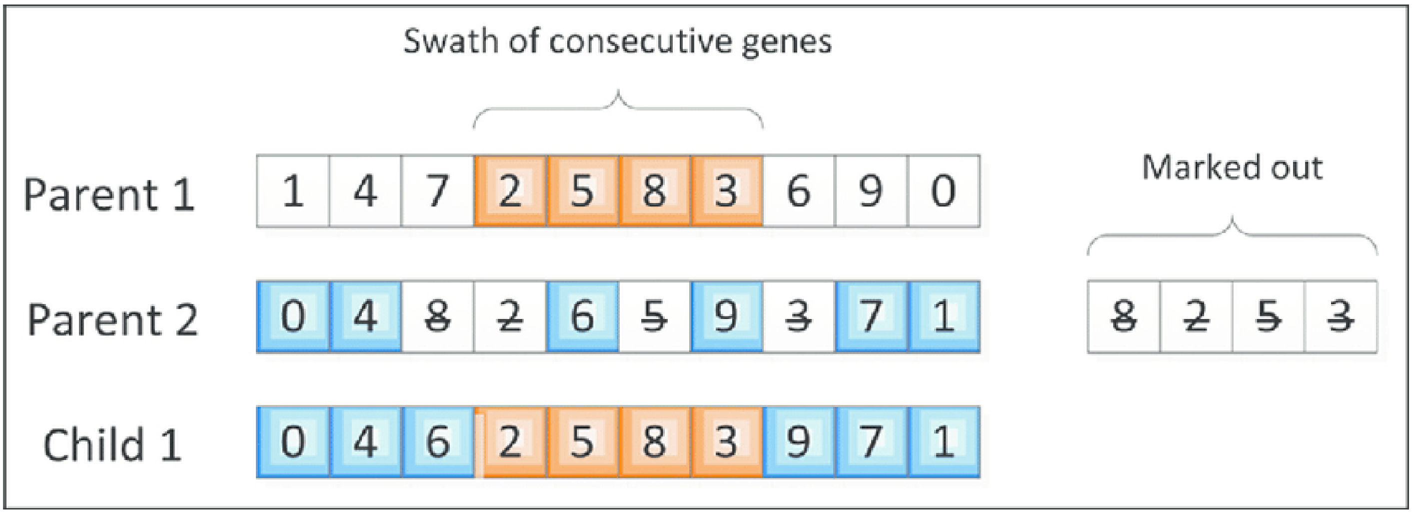

Figure 2: Demonstration of Order One Crossover

2 Results and Report

We will try to solve 25-Queen and compare each method. I will run the program two times for each method and calculate the average of the generation of the responses. The lower this number the better is the method. First, we try different population sizes with each method and with crossover rate of 80% and mutation rate of 20%. Then, when we find the best population size, we try to adjust the other two parameters.

There are 8 possible combination of the methods that we described. However, the methods with a selection method of Roulette wheel, would not generate solutions for any N bigger than 20. As a result, for the 25-Queen problem, we Have 4 combinations of methods that start with Tournament selection.

In the Figure 3 we have depicted the effect of population size on the generation of the final answer for each method.

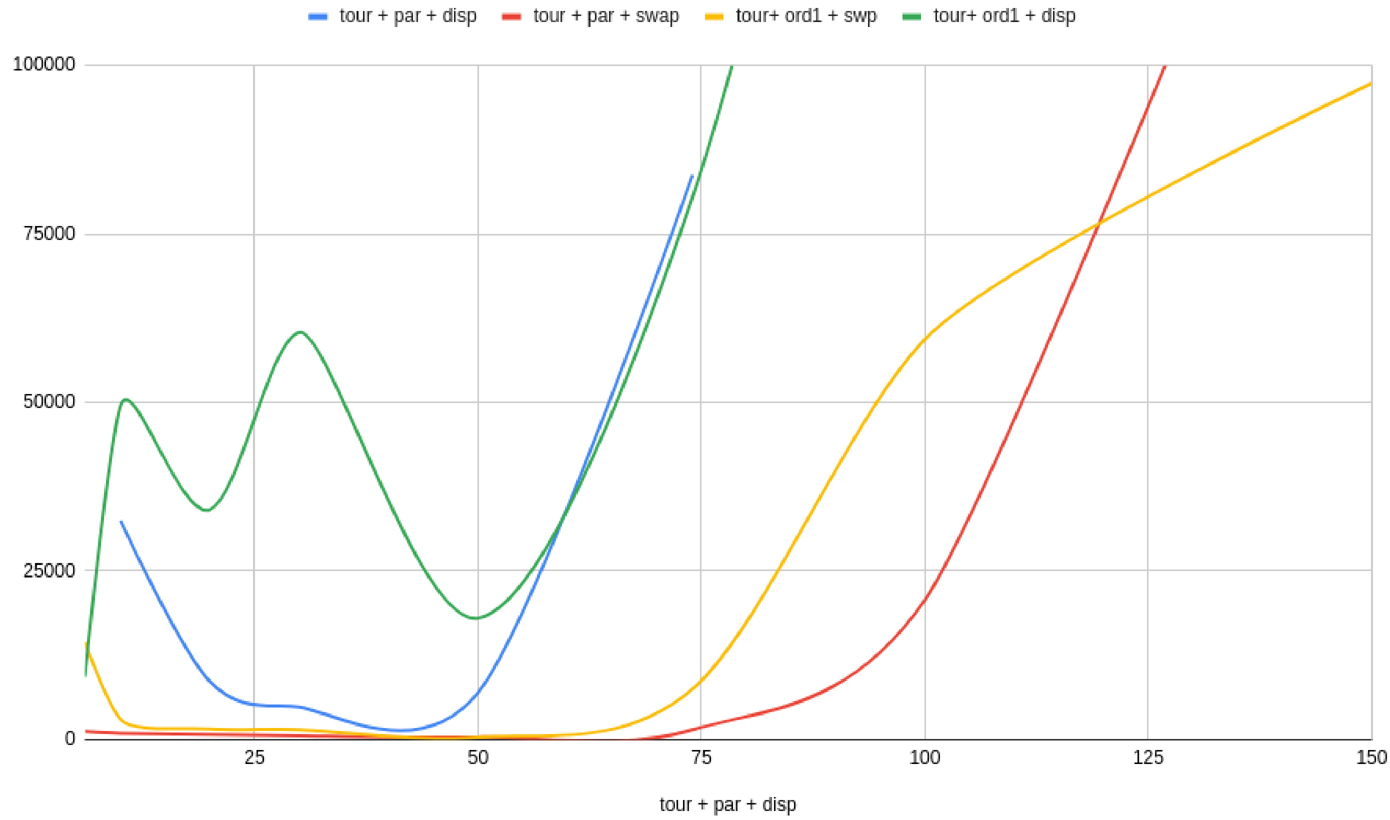

Figure 3: Effect of Population size on each of the 4 methods

We can see that the combination of Tournament Selection, Partially Mapped Crossover and Swap Mutation with a population of 50 gives us a slightly better result than the other methods. This slight advantage is more clear with larger N-Queen problems.

Next, we try to adjust the crossover and mutation rate with the best combination and the optimum population size that we found.

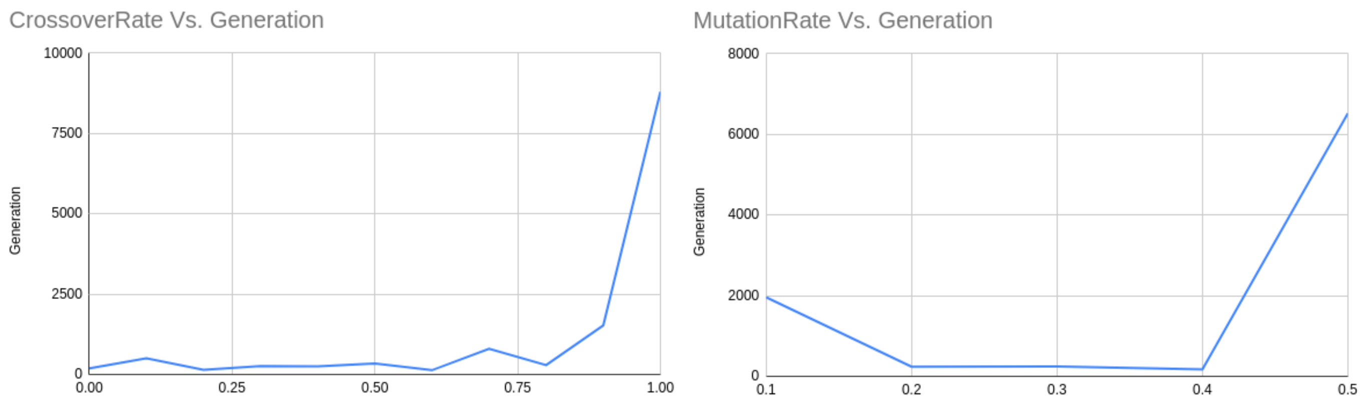

Figure 4: Crossover and Mutation Rate effect on the final answer's generation

From Figure 4 we find out that on average, the best Crossover rate is about 60% and the best mutation rate is about 40%.

Finally, we have included the answers to some big N-Queen problems like 500-Queen and 1000-Queen.

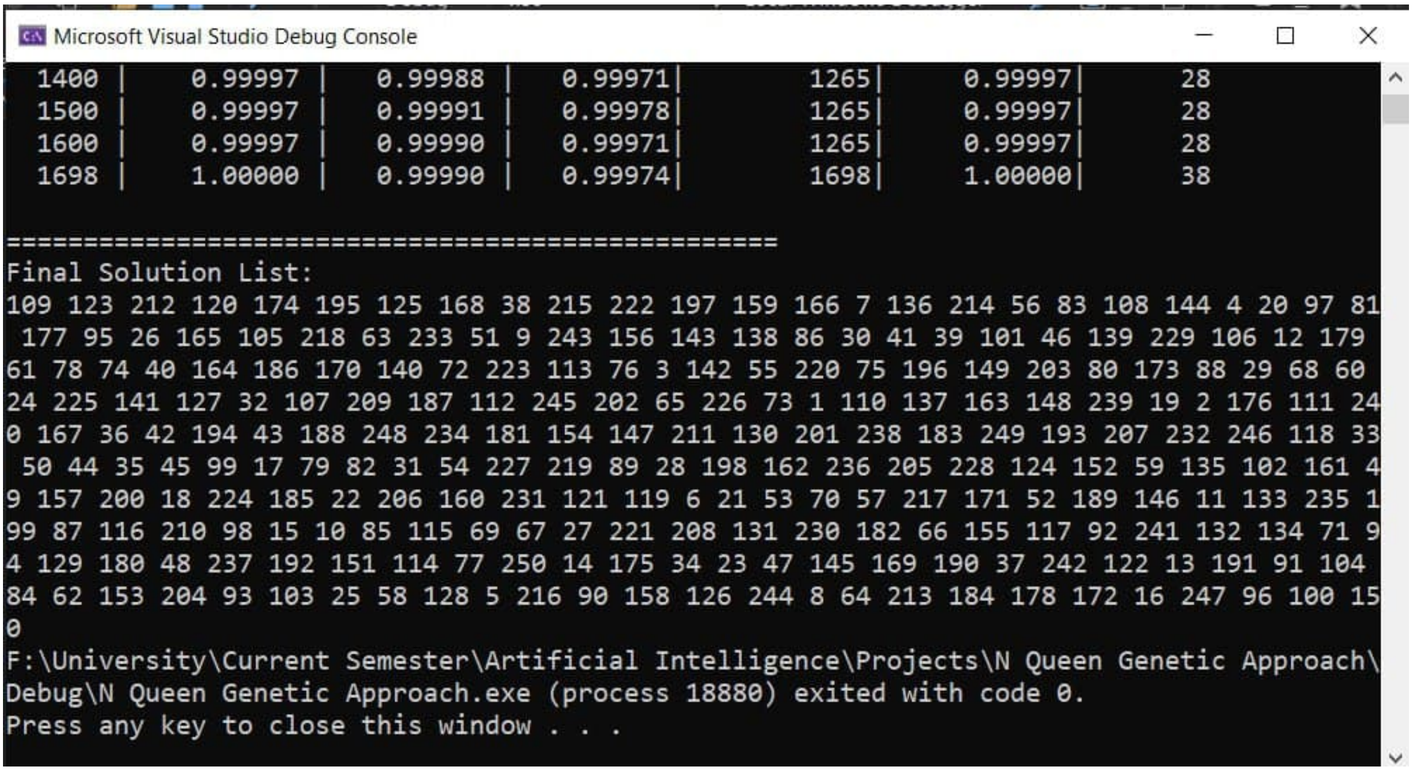

Figure 5: Solution to the 250-Queen Problem

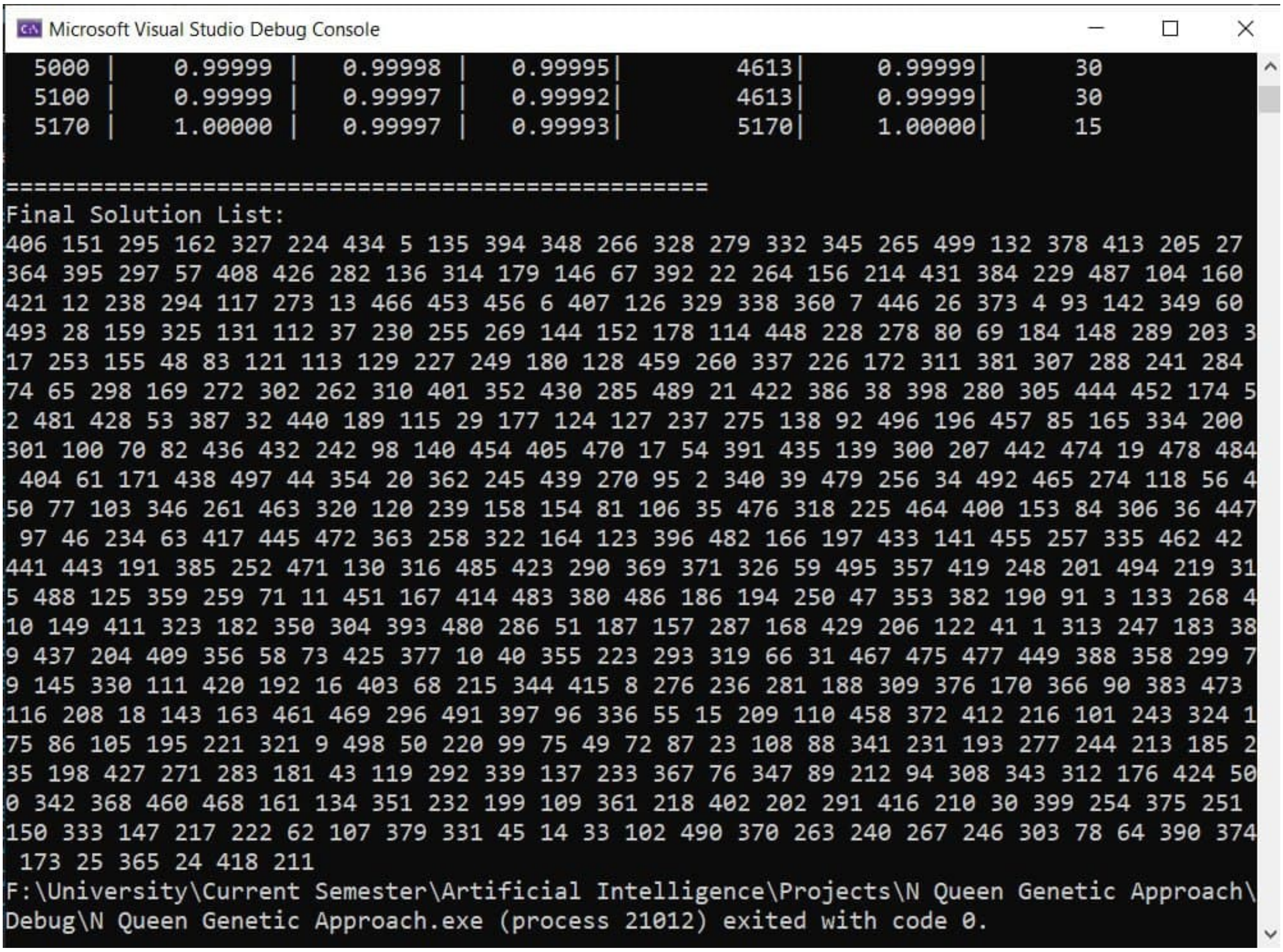

Figure 6: Solution to the 500-Queen Problem

```
 17600 |    1.00000 |   0.99999 |   0.99998|       12064|     1.00000|      46
 17700 |    1.00000 |   0.99999 |   0.99998|       12064|     1.00000|      46
 17800 |    1.00000 |   0.99999 |   0.99998|       12064|     1.00000|      46
 17809 |    1.00000 |   0.99999 |   0.99999|       17809|     1.00000|      44

==================================================
Final Solution List:
711 343 662 444 787 755 695 485 312 770 635 665 536 220 342 374 874 683 790 630 146 382 947 457 794 285 617 145 597 903 879 789
 18 510 608 100 26 92 747 470 525 76 318 171 940 881 640 306 771 606 123 750 132 778 310 335 414 68 782 324 352 397 403 761 452
 456 893 496 301 904 142 591 189 571 502 582 448 166 516 208 776 404 713 540 277 653 583 71 385 307 424 441 580 256 538 216 375
 637 584 469 565 927 334 265 236 508 832 16 242 276 631 833 627 257 784 163 864 148 590 34 822 241 367 480 643 651 178 492 380
453 638 883 478 344 951 425 566 69 25 514 407 51 728 262 247 38 43 686 364 468 602 286 779 2 400 843 418 300 700 574 251 443 83
4 758 550 974 621 757 84 447 7 45 419 544 977 607 515 706 732 684 13 297 103 724 90 941 309 229 193 632 30 595 882 556 929 975
689 821 930 152 522 967 748 192 224 431 943 87 774 726 499 426 218 610 308 322 353 488 73 521 966 964 859 89 459 361 849 303 49
1 465 531 644 357 673 812 990 384 253 857 549 517 956 995 937 323 458 49 445 628 466 185 931 23 222 718 117 351 902 681 150 274
 669 371 377 877 494 611 860 360 316 164 670 741 119 863 118 8 915 240 815 948 840 674 383 532 211 824 555 769 65 710 493 289 6
93 715 753 626 567 128 675 66 781 505 752 337 926 273 298 143 313 64 182 110 477 687 12 839 512 122 296 804 633 255 509 895 889
 954 270 861 616 731 896 935 618 603 957 868 259 83 423 704 498 982 288 3 575 107 85 911 796 252 429 847 950 101 534 577 887 56
 167 666 772 788 381 797 74 546 199 559 919 393 212 663 238 856 174 730 985 744 230 854 449 697 917 878 716 228 59 810 513 366
325 828 9 331 302 897 999 401 890 482 427 372 156 78 851 278 916 10 304 321 127 217 671 378 130 329 124 853 244 862 709 258 884
 913 386 998 600 867 86 743 196 246 587 227 901 529 46 134 387 24 850 564 545 869 932 395 691 898 373 507 188 842 47 727 120 70
7 953 921 77 187 455 961 690 116 141 287 20 37 807 160 54 997 206 803 245 785 875 994 231 645 829 195 412 280 139 720 479 415 2
10 17 551 197 63 165 578 102 239 965 659 962 844 615 654 712 133 437 677 93 979 678 836 699 4 866 942 692 317 22 588 151 996 43
9 819 180 33 541 946 848 714 349 461 738 838 855 518 557 495 680 733 696 305 48 543 108 811 416 31 267 749 328 701 1000 202 910
 430 52 925 35 388 939 406 184 275 768 762 129 356 144 646 736 483 537 958 428 299 435 232 390 504 80 200 36 332 476 827 219 80
1 149 81 952 135 405 315 825 155 899 992 976 968 355 945 845 852 993 650 987 464 552 756 460 121 157 347 773 865 703 399 233 26
1 158 394 938 269 442 735 973 94 629 420 369 802 682 605 213 264 609 161 234 19 58 813 154 739 345 79 983 422 99 984 933 826 75
 702 341 226 539 642 136 503 799 723 191 620 201 42 885 816 888 908 944 576 440 379 527 198 837 823 688 96 988 649 745 40 203 6
24 284 792 960 147 179 594 50 450 293 263 506 32 894 292 350 667 279 159 959 766 327 186 949 914 892 138 60 817 391 553 589 907
 125 181 501 760 82 721 365 586 91 679 72 311 524 923 906 970 55 809 175 777 358 105 598 981 29 11 795 767 876 969 97 535 243 4
54 326 806 271 15 467 282 717 204 67 661 44 411 484 604 27 333 497 295 593 396 905 871 248 489 652 814 413 991 989 560 599 592
528 886 820 462 648 972 880 95 660 657 314 250 266 283 533 759 775 112 562 955 62 891 98 433 634 936 368 725 924 694 249 104 62
3 339 754 765 636 173 221 805 742 346 764 705 260 487 392 818 205 113 596 526 831 88 115 619 57 111 486 53 548 563 858 658 281
223 490 446 294 168 170 986 900 409 520 172 783 472 434 376 746 980 421 61 585 928 320 909 613 28 519 183 359 763 6 572 471 473
 290 685 417 668 436 511 215 581 830 362 920 272 841 209 109 786 114 131 500 729 873 912 398 963 336 601 410 934 438 708 978 21
 664 523 1 570 846 579 474 291 791 176 481 568 793 225 106 162 625 561 39 330 922 971 835 639 340 475 207 177 612 338 153 389 7
80 719 800 41 354 798 140 451 542 370 237 641 463 408 194 126 214 918 647 569 740 614 558 14 737 432 5 348 530 254 656 137 235
```

Figure 7: Solution to the 1000-Queen Problem