Development of a Neural Net Model for the Assignment Problem

Philip Paul Moss

Western Michigan University

Follow this and additional works at: https://scholarworks.wmich.edu/masters_theses

Recommended Citation
https://scholarworks.wmich.edu/masters_theses/1090
DEVELOPMENT OF A NEURAL NET MODEL FOR THE ASSIGNMENT PROBLEM

by

Philip Paul Moss

A Thesis Submitted to the Faculty of The Graduate College in partial fulfillment of the requirements for the Degree of Master of Science Department of Computer Science

Western Michigan University Kalamazoo, Michigan June 1990
DEVELOPMENT OF A NEURAL NET MODEL FOR THE ASSIGNMENT PROBLEM

Philip Paul Moss, M.S.
Western Michigan University, 1990

This thesis traces the development of a neural net model which will solve the assignment problem. In addition, the neural net model was compared for optimal performance with four other algorithms: two greedy methods and two well-known solutions, Vogel's Approximation Method (VAM) and the Hungarian Method. Assignments were made to randomly generated problems using each method and the results were compared.

The findings from this testing indicated that: (a) the assignments selected by the neural net model were closer to optimal than the two greedy methods and VAM, and (b) the assignments selected by the neural net were not optimal. It was concluded that the neural net model, although not optimal, could have useful applications.
ACKNOWLEDGEMENTS

I wish to extend sincere appreciation to my advisor and committee chairperson, Dr. Robert Trenary, for his encouragement, assistance, direction, and support throughout my entire course of study; and to my committee members, Dr. Ben Pinkowski and Dr. Donna Kaminski, for their advice and guidance.

Lastly, my deepest thanks and appreciation are extended to my wife, JoEllen Sandor, for her assistance in proofreading and for providing the love, support, and encouragement needed to complete this thesis.

Philip Paul Moss
INFORMATION TO USERS

The most advanced technology has been used to photograph and reproduce this manuscript from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.
Development of a neural net model for the assignment problem

Moss, Philip Paul, M.S.
Western Michigan University, 1990
TABLE OF CONTENTS

ACKNOWLEDGEMENTS .................................................................................................................... ii
LIST OF TABLES .................................................................................................................................... vi
LIST OF FIGURES .................................................................................................................................. viii

CHAPTER

I. INTRODUCTION .................................................................................................................. 1

II. NEURAL NETWORKS ........................................................................................................ 3
    Introduction .............................................................................................................................. 3
    General Characteristics ........................................................................................................ 3
    Processing Elements ............................................................................................................ 4
    Activation State .................................................................................................................... 5
    Connectivity Between Processing Elements .............................................................. 6
    Output Function ..................................................................................................................... 6
    Propagation Rule ................................................................................................................... 7
    Activation Rule ..................................................................................................................... 8
    Additional Characteristics of a Neural Network ...................................................... 9
    Examples of Neural Networks .......................................................................................... 9
    Summary of Notation .......................................................................................................... 11

III. THE ASSIGNMENT PROBLEM ....................................................................................... 14
    Introduction and Example Problem ............................................................................... 14
    Representations of the Assignment Problem .......................................................... 16

IV. THE ASSIGNMENT PROBLEM AS A NEURAL NETWORK ................................ 19
    Introduction ........................................................................................................................ 19

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
# Table of Contents—Continued

## CHAPTER

<table>
<thead>
<tr>
<th>The Initial Neural Networks</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addition of Hidden Layers to the Neural Network</td>
<td>22</td>
</tr>
<tr>
<td>Experiments With the Neural Network Algorithm</td>
<td>26</td>
</tr>
<tr>
<td>Greedy Method #1</td>
<td>26</td>
</tr>
<tr>
<td>Greedy Method #2</td>
<td>26</td>
</tr>
<tr>
<td>Test Results</td>
<td>28</td>
</tr>
<tr>
<td>The Neural Network Algorithm</td>
<td>30</td>
</tr>
</tbody>
</table>

## V. OTHER APPROACHES TO THE ASSIGNMENT PROBLEM

| Introduction | 33 |
| Vogel's Approximation Method (VAM) | 33 |
| Branch-and-Bound Techniques | 37 |
| The Hungarian Method | 41 |
| Initial Reduction | 42 |
| Initial Assignment | 44 |
| Labelling | 46 |
| Further Reduction | 48 |
| Breakthrough | 50 |
| Additional Comments | 53 |

## VI. TEST RESULTS

| Introduction | 54 |
| Test #1 - Basic Performance | 54 |
| Tests #2 and #3 - Changes in Matrix Weights | 55 |
| Tests #4 and #5 - Matrix Size Changes | 57 |
Table of Contents—Continued

CHAPTER

Interpretation of Test Results.......................................................... 59

VII. CONCLUSION............................................................................. 61

Miscellaneous Comments............................................................... 61

Similarities and Differences.......................................................... 61

Enhancements............................................................................... 62

APPENDICES

A. Source Code............................................................................... 63

B. Test Results............................................................................... 85

BIBLIOGRAPHY.............................................................................. 115

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
LIST OF TABLES

1. Sample Librarian Problem .......................................................... 15
2. The Assignment Problem Represented by a Matrix .......................... 17
3. The Matrix Representation for a Selected Assignment ....................... 18
4. Comparison of Neural Network Method With Two Greedy Methods ....... 28
5. Neural Network Result on Matrices With Known Maximums .............. 29
6. Results With Various Multipliers on Matrices With Known Solutions (50 Trials) ................................................... 29
7. Results With Various Multipliers on Matrices With Unknown Solutions (50 Trials) ................................................... 30
8. Original Matrix With Initial Penalties - PR, & PCj ............................ 34
9. Matrix After First Reduction (Row 5,Column 2) With New Penalties - PR, & PCj ................................................... 34
10. Matrix After Second Reduction (Row 1,Column 5) With New Penalties - PR, & PCj .................................................. 35
11. Matrix After Third Reduction (Row 2,Column 6) With New Penalties - PR, & PCj .................................................. 35
12. Matrix After Fourth Reduction (Row 4,Column 4) With New Penalties - PR, & PCj .................................................. 36
13. Di for the Example Matrix ......................................................... 37
14. Test Results 13 X 13 Matrix With Weights 1 - 13 ............................ 54
15. Percentage Comparison to the Hungarian Method .......................... 55
16. Test Results 13 X 13 Matrix With Weights 1 - 26 ............................ 56
17. Percentage Comparison to the Hungarian Method .......................... 56
18. Number of Correct Solutions Found Compared With Hungarian Method ................................................. 56
19. Test Results 13 X 13 Matrix With Weights 1 - 7 ............................ 57
List of Tables—Continued

20. Percentage Comparison to the Hungarian Method ......................................................... 57
21. Number of Correct Solutions Found Compared With Hungarian Method .......................................................... 57
22. Test Results 20 X 20 Matrix With Weights 1 - 20 ............................................................ 58
23. Percentage Comparison to the Hungarian Method ........................................................ 58
24. Test Results 8 X 8 Matrix With Weights 1 - 8 ................................................................. 59
LIST OF FIGURES

1. A General Formalism for a Neural Network......................... 4
2. Three Common Output Functions................................................. 7
3. Sigmoid Activation Rule.............................................................. 8
4. Two Neural Networks Which Reproduce the Exclusive-Or Truth Table..... 10
5. A Generalized Perceptron Which Uses the Learning Rule Discussed in the Text... 12
6. The Assignment Problem Represented by a Graph........................... 16
7. Initial Neural Network Representation of the Assignment Problem........... 21
8. Modified Neural Network - Connection From Input PEi to Output PEj Added.......................... 22
9. Neural Network With Excitatory Connections Removed.......................... 23
10. Neural Network With Hidden Layers Added...................................... 24
11. Final Neural Network Model for the Assignment Problem...................... 25
12. Pseudo-Code for Greedy Method #1............................................. 27
13. Pseudo-Code for Greedy Method #2............................................. 27
14. First Step in Branch-and-Bound Technique................................. 38
15. Branching From 1,1.................................................................. 39
16. Branching From 2,1.................................................................. 39
17. Branching From 3,1.................................................................. 40
18. Branching From 4,1.................................................................. 40
19. Branching From 5,1.................................................................. 40
20. Branching From 6,1.................................................................. 40
21. Branching From Node 3,2 Which Leads to the Final Assignment................. 41

viii

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
List of Figures—Continued

22. The Hungarian Method's Solution to the Example Problem ........................................... 42
23. Initial Matrix Showing Row Minimums ........................................................................ 43
24. Row Reduced Matrix Showing Column Minimums ....................................................... 45
25. Fully Reduced Matrix With Initial Assignments .......................................................... 47
26. Fully Reduced Matrix With Initial Assignments and Labelling ................................... 49
27. Matrix After First Reduction With New Assignments and Labelling ......................... 51
28. Matrix After Second Reduction With New Labelling and Final Assignment .......... 52
CHAPTER I

INTRODUCTION

The Assignment Problem gets its name from applications in which we wish to assign "workers" to "tasks." We assume each worker must be assigned to a single task and each task must be performed by a single worker. Many personnel assignment problems fit the assumptions and structure of this problem.

There are many examples of the assignment problem occurring in everyday situations. Some of these could be: (a) a factory foreman assigns the workers to a job, (b) a baseball manager fills out his line-up card, (c) a laboratory manager assigns the technologists to a test area, or (d) a swimming coach chooses the team members to swim each leg in the medley relay. Despite the differences in each of these examples, they have some common features:

1. The goal of each assignment is to minimize or maximize the solution. The laboratory manager and factory foreman are attempting to maximize production while holding errors to an acceptable limit, the baseball manager attempts to win games by maximizing offensive production and minimizing defensive liabilities, and the swimming coach attempts to win the relay by minimizing the total time.

2. Each assignment involves some attribute or attributes which can be ranked or quantified and used by the assigner for his selections. The manager knows batting averages, fielding averages, and skill at certain positions and can make his choices with this knowledge or the laboratory manager knows the strengths and weaknesses of his employees and makes assignments using this knowledge.

3. Each assignment involves a certain amount of gain or loss which must be evaluated. If the baseball manager chooses to play his best nine hitters, there might not be adequate
defensive players; if the manager selects his best defensive team, the team offense may be inadequate; and if the swim coach allows his fastest free style swimmer to swim the free style leg of the relay, the butterfly leg and the overall time may be poorer.

This paper attempts to develop a neural net model and algorithm which will optimize assignments. The neural net approach was chosen because the features described above seem to lend themselves to a neural net solution. This paper consists of five major topics. Chapter II presents an overview of neural nets and develops a notation which will be used throughout. Chapter III presents the assignment problem and different models which can be used to represent it. Chapter IV discusses the steps taken and problems encountered as a neural net model and algorithm was developed for the assignment problem. Chapter V investigates alternate methods that can be used to approach the assignment problem. Chapter VI presents a series of test runs on IBM-PC comparing the results obtained using the neural net approach with two of the methods presented in Chapter V. A final chapter discusses the similarities of the neural net approach and other methods, possible improvements to the neural net approach, and shortcomings of the neural net approach. Appendices contain the source code for all programs developed and complete raw data from the various tests performed.
CHAPTER II

NEURAL NETWORKS

Introduction

A neural network assumes processing takes place through the interaction of a number of simple processing elements, each sending excitatory and/or inhibitory signals to other processing elements. This section will describe the general characteristics of a neural network, present example neural networks which illustrate these characteristics, and develop a notation which can be used to describe a specific neural network.

General Characteristics

The six major characteristics of a neural net (Rumelhart, Hinton, & McClelland 1986) are: (1) a set of processing elements, (2) an activation state of a processing element, (3) connectivity between processing elements, (4) an output function for each processing element, (5) a propagation rule which defines how other processing elements affect individual processing elements, and (6) an activation rule which determines the next activation state of a processing element.

The principal elements of a neural network are illustrated in Figure 1. Processing elements are represented as circles. Each processing element (PEj) has an activation state denoted as aj(t) indicating the state of PEj at time t. This activation is passed through an output function to produce an output value denoted as Oj(t). Each connection between PE's has a strength or weight assigned to it denoted as wt, which determines the effect the PEi has on PEj. The propagation rule is expressed as netj, a function of the outputs of the other PE's
and the weight of connections to PE_j. The activation rule is shown as a_j(t+1) which is a function of the current activation a_j(t) and the net input to the PE net_j.

**Processing Elements**

All neural network models begin with a set of processing elements. Defining what these elements represent is the first stage in specifying a model. The processing element is a small feature-like entity, while the combination of all the processing elements becomes the meaningful entity.

The processing elements are numbered arbitrarily and we refer to the ith element as PE_i.
All processing is carried out locally in the processing elements; there is no overseer or executive processing element which maintains a global view of the state of the network. A processing element receives input from other connected processing elements, calculates an output value, and sends that output value to other processing elements to which it is connected.

The processing elements change activation state in parallel, either synchronously or asynchronously. Within a given neural network, there can be three basic types of processing elements: (1) input, (2) output, and (3) hidden. Input processing elements receive their signals from an external source, i.e., net, is not produced from other processing elements. Output processing elements send their outputs to an external location, i.e., o, is not connected to other processing elements. Hidden processing elements are not visible to external inputs or outputs.

**Activation State**

The collective activations of the N processing elements in the neural network represent the state of the model at time t. This state can be specified by a vector of N real numbers a(t) with the activation state of each processing element at time t being represented by one of the entries in the vector. The activation of processing element i at time t is represented by a_i(t).

Different neural network models allow the activation state to take different values. The values may be continuous or discrete. In neural network models which allow continuous activation values, these values may be unbounded—values may assume any real value—or bounded—values can assume any real value in a given range, e.g., [-1,1]. In neural network models which allow discrete activation values, these values may be binary, e.g., [0,1] or [-1,1] or these values may be from a small set of possible values, e.g., [-1,0,1].
Connectivity Between Processing Elements

Processing elements influence each other through connections and it is the pattern and the strength (weight) of these connections which determines the neural network's response to a given input.

A matrix provides a convenient representation of the connectivity of a neural network. In this case, the connectivity may be represented by an N X N weight matrix $W$ in which entry $w_{ij}$ represents the strength and excitatory/inhibitory nature of the connection between processing element $PE_j$ and processing element $PE_i$. The weight $w_{ij}$ is positive if the connection is excitatory and $w_{ij}$ is negative if the connection is inhibitory. The absolute value of $w_{ij}$ determines the strength of the connection.

A more general connectivity is often required because all inputs are not equivalent. In this case the connectivity is represented by multiple matrices $W_w$ one for each type of input, and the totals from each of these matrices are combined for the total input to a processing element.

The connectivity of a neural network plays a major role in determining what each processing element represents. Connectivity determines if top-down (elements at level $i$ do not affect the activity of elements at levels lower than $i$), bottom-up (elements at level $i$ do not affect the activity of elements at levels higher than $i$), or interactive (any element can affect any other element) processing is correct. Connectivity determines levels of hierarchy which affect the degree of parallelism which can be obtained. In many cases an extremely general connectivity is required.

Output Function

Processing elements in a neural network model interact by transmitting signals to
neighboring processing elements. The strength of this influence is determined by a processing element's activation. Each processing element PEi has an output function f_i(a_i(t)) associated with it. This function maps the current activation state to an output signal, i.e., o_i = f_i(a_i(t)). In vector notation, the output values are represented as o(t). Various output functions are used in neural networks; three common output functions are: (1) identity function f(x) = x, (2) threshold function in which a processing element has no effect on other processing elements until its activation state exceeds a certain value, e.g., o_i = 1 if a_i > 0 else o_i = 0, and (3) a probabilistic function in which the value of o_i depends in some probabilistic way on the value of a_i. Examples of a continuous function, a threshold function, and a combination threshold/continuous function are illustrated in Figure 2.

![Graphs of Continuous, Threshold, and Combination Functions](image)

**Figure 2. Three Common Output Functions.**

**Propagation Rule**

In a neural network a propagation rule is required to show how other processing elements affect a particular processing element. A propagation rule combines the outputs of
processing elements and the connectivity matrix to produce a net input (for each type of input) to the subject processing element. The notation used for propagation rules is $\text{net}_{i,j} = f(\{o_{i,j}\},\{w_{i,j}\})$ for inputs of type $j$ to processing element $i$. If only one connectivity is involved, the rule is written $\text{net}_{i,j} = f(\{o_{i,j}\},\{w_{i,j}\})$.

Propagation rules are usually quite simple. For example, with excitatory and inhibitory connections to a processing element the rule is commonly that net excitatory input is the weighted sum of the individual excitatory inputs and the net inhibitory input is the weighted sum of the individual inhibitory inputs. Other propagation rules are possible; a sigma-pi rule is the most common. In this case the products of the outputs and the weights are used instead of the sum of the inputs. In vector notation, the notation used is $\text{net}_j(t)$ which represents the net input vector of type $j$ inputs.

![Sigmoid Activation Rule](image)

**Figure 3. Sigmoid Activation Rule.**

**Activation Rule**

The activation rule is used to determine the next state of a processing element. This rule
uses the current activation state $a_i(t)$ and the net inputs $net_i$ to determine the next activation state $a_i(t+1)$. The activation rule is then the function $F$ such that $a_i(t+1) = F(a_i(t)), [net_i(t)]).$ This function is normally deterministic, i.e., $a_i(t+1)$ will assume the value 1 if $net_i(t) > 0$ and is 0 otherwise. The activation rule may be stochastic or probabilistic. Activation values can also be considered a decaying value so that a processing element receiving no external input will degrade and not go directly to zero. If $a_i(t)$ is assumed to have continuous values, a sigmoid activation rule (shown in Figure 3), which is a differentiable function, is often used.

**Additional Characteristics of a Neural Network**

Any neural network model must also specify the environment in which it operates. The environment is determined by the task the neural network is to perform and can be defined in terms of inputs to and outputs from the various processing elements. The most common environment is one in which any possible input pattern may be impinging on the input processing elements.

One additional aspect found in many neural networks is a learning rule whereby the neural network is changed dynamically in order to improve task performance. A learning rule is usually one of three types: (1) connectivity is added, (2) connectivity is eliminated, or (3) weight of connectivity is changed. Various learning rules involving the third type have been developed by Hebb (1949), Widrow and Hoff (1960), and Grossberg (1976) among others.

**Examples of Neural Networks**

Three sample neural networks are shown in Figures 4 and 5. The first two examples (shown in Figure 4) demonstrate two different neural networks (Rumelhart, Hinton, & Williams 1986) which compute the exclusive-or function. Exclusive-or has two inputs and one output. The output is on if the two inputs have different activations and is off when the two
inputs have the same activation. These examples demonstrate that the number of processing elements may vary, that processing elements may have varying thresholds (the hidden layer in example #2 has a threshold of 1, while other elements have a threshold of 0), and that weights may be negative or positive.

![Figure 4. Two Neural Networks Which Reproduce the Exclusive-Or Truth Table](image)

The third example (shown in Figure 5) is the Perceptron developed by Rosenblatt (1962). This neural network presents a simple learning rule and uses processing elements, called Threshold Logic Units (TLU), which use the threshold activation function. The Perceptron is presented a set of inputs and desired outputs. Arbitrary weights are assigned and the following learning procedure is applied. If an incorrect output is obtained at an output
processing element and this output is a zero, then the weights on all connections to this processing element from other active processing elements is increased by delta, a small weight change. If an incorrect output is obtained at an output processing element and this output is one, then the weights on all connections to this processing element from other active processing elements are decreased by delta. The Perceptron continues in this manner until a correct connectivity is obtained (if possible). Rosenblatt showed that a two layer Perceptron could produce proper weighting for any of the \(2^N\) classifications of \(N\) binary inputs, but \(2^N\) processing elements were required for the completely general case. Subsequent study by Minsky and Papert (1969) has shown that if TLU's are used and a multi-layer network is allowed, then about \(3N\) processing elements and \(\log_2 N\) layers will produce the desired classification.

**Summary of Notation**

The notation which is used for the elements in a neural net is summarized below.

<table>
<thead>
<tr>
<th>Processing Element</th>
<th>PE, the (i)th processing element</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activation State</td>
<td>(a_i(t)) the state of PE, at time (t)</td>
</tr>
<tr>
<td>Connectivity</td>
<td>(w_{ij}) the connection between PE, and PE, with a negative weight indicating inhibition, a positive weight indicating excitation and the absolute value of (w_{ij}) indicating strength of the connectivity.</td>
</tr>
<tr>
<td>Output</td>
<td>(o_i(t)) where (o_i(t) = f(a_i(t))) the output value of processing element (i) at time (t) is a function of the activation value of processing element (i) at time (t).</td>
</tr>
<tr>
<td>Propagation Rule</td>
<td>(net_i(t)) where (net_i(t) = f([o_i(t)], w_{ij})) the net input to processing element (i) at time (t) is a function of the outputs of other processing elements (j) at time (t) and the connectivity between processing element</td>
</tr>
</tbody>
</table>

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
i and other processing elements \(j\).

Activation Rule

\[ a_i(t+1) = F(a_i(t), \{\text{net}_i(t)\}) \]

The activation state of processing element \(i\) at time \(t+1\) is a function of the activation state of processing element \(i\) at time \(t\) and the net input to processing element \(i\) at time \(t\).

\[ \text{net}_j = \sum w_{j,i} \cdot o_j \]

\[ \text{PE}_i = \text{processing element} \]

\[ o_j = \text{threshold function on } a_j(t) \]

**Figure 5.** A Generalized Perceptron Which Uses the Learning Rule Described in the Text.

Neural networks can be developed which perform a variety of tasks, including problems in information retrieval, motor control, perception, and pattern recognition, all within a parallelized distribution framework. As mentioned in the introductory chapter, the assignment problem presents many features which lend themselves to a distributed parallel approach and thus a neural network approach. The following chapter will present a general
overview of the assignment problem.
CHAPTER III

THE ASSIGNMENT PROBLEM

Introduction and Example Problem

In structure, the assignment problem is a special subclassification of the Transportation Problem (Ignizio 1982, pp. 278-317). The Transportation Problem (also known as the Distribution Problem) consists of "sources" (points of supply), "sinks" (points of demand), paths from source to sink, constraints on the amount of goods which may be moved along a given path, and constraints on the amounts available at sources and desired at sinks. The objective of the transportation problem is minimization of cost while satisfying the demand at the sinks within the path constraints. In the assignment problem, each worker can be considered a source of one unit, each task can be considered a sink demanding one unit, and the cost is the weight of worker i performing task j. The assignment problem is often embedded within larger problems and the assignment problem (Taha 1975, pp. 305-316) has been used to produce approximations with the Traveling Salesperson Problem. The assignment problem can be treated as a transportation problem for solution, but special techniques have been developed to handle the particular characteristics of the assignment problem.

The objective of the assignment problem is to minimize (or maximize) the total costs of the assignments. These costs may be in dollars required to complete the tasks, time required to complete the tasks, minimum error to complete the tasks, or many other possibilities.

A problem presented by Tank and Hopfield (1987) is the Librarian Problem. Assistant librarians can reshelve books at various rates depending upon the subject area they are reshelving. The head librarian must try to maximize the number of books that can be
reshelved in unit time. Assumptions are: (a) there are N assistant librarians and N areas to reshelve, (b) each assistant will work in only one area, and (c) each area will be worked in by only one assistant. An example problem is shown in Table 1.

Table 1

<table>
<thead>
<tr>
<th>Student</th>
<th>History</th>
<th>Math</th>
<th>Science</th>
<th>Art</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Smith</td>
<td>31</td>
<td>32</td>
<td>34</td>
<td>29</td>
</tr>
<tr>
<td>Mary Jones</td>
<td>36</td>
<td>30</td>
<td>37</td>
<td>33</td>
</tr>
<tr>
<td>Harry Schmidt</td>
<td>34</td>
<td>34</td>
<td>37</td>
<td>35</td>
</tr>
<tr>
<td>Sparky Anderson</td>
<td>30</td>
<td>33</td>
<td>31</td>
<td>28</td>
</tr>
</tbody>
</table>

In this example, the optimal result is obtained with the following assignment: (a) John Smith to Science, (b) Mary Jones to History, (c) Harry Schmidt to Art, and (d) Sparky Anderson to Math. The total rate is 138 books per unit time. Note that if a "greedy" method, in this case assigning the largest reshelving rates first, is used, then the total rate is reduced. For example, if Mary Jones is assigned to Science the maximum rate is 135 books per unit time; while if Harry Schmidt is assigned to Science the maximum rate is 136 books per unit time.

The assignment problem is a problem in combinatorics and people generally underestimate the complexity. In the above example with four assistants and four areas, there are twenty-four possible solutions. Enumeration is not a viable option, because the number of enumerations is N!; with 10 assistants and ten subject areas the number of possible
Representations of the Assignment Problem

The Assignment Problem can be represented by two different models, a graph model and a matrix model, and can be expressed mathematically (Ignizio 1982, pp. 321-323). To represent the above example as a graph, let each assistant and each area be a vertex. Join each assistant and area with an edge and let the reshelving rates be the cost of the edges. The result is a bi-partite graph and is shown in Figure 6.

![Graph Model](image)

Figure 6. The Assignment Problem Represented by a Graph.

An assignment can be represented by removal of all edges not in the assignment. In graph theory, a matching is a partial graph in which every vertex has degree less than or equal to one. A complete or perfect matching is a graph where all vertices have degree one. For this reason the Assignment Problem is also known as the bi-partite matching problem.
(Murty 1976).

The graph is identical to that developed for the transportation problem except (a) availability at each source is one, (b) demand at each sink is one, and (c) the number of sources and sinks is identical. In this example, John Smith, Mary Jones, Harry Schmidt, and Sparky Anderson are sources while History, Math, Science, and Art are sinks.

When the assignment problem is represented in a matrix model, the assistant librarians represent rows, the subject areas represent columns, and the rates of reshelving are represented as the matrix values at the assistant/area intersections. The librarian problem example is shown in Table 2.

Table 2

<table>
<thead>
<tr>
<th>Assistant Librarian = row</th>
<th>Subject Area = column</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rate of Reshelving</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>32</td>
</tr>
<tr>
<td>36</td>
<td>30</td>
</tr>
<tr>
<td>34</td>
<td>34</td>
</tr>
<tr>
<td>30</td>
<td>33</td>
</tr>
</tbody>
</table>

An assignment is represented by a second N X N matrix with a 1 at entry i,j if assistant i is assigned to area j and a 0 otherwise. The solution for this example is shown in the assignment matrix in Table 3.

The total weight or rate is obtained by multiplying entries in the original matrix and the assignment matrix that have identical subscripts and adding these results, i.e., the total rate
\[ Z = \sum \sum c_{ij} \cdot a_{ij} \] where \( c_{ij} \) is the rate assistant \( i \) shelves area \( j \) and \( a_{ij} \) is 1 if assistant \( i \) is assigned to area \( j \) and 0 otherwise.

Table 3

<table>
<thead>
<tr>
<th>Assistant Librarian = row</th>
<th>Subject Area = column</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignment Matrix</td>
<td></td>
</tr>
<tr>
<td>0 0 1 0</td>
<td></td>
</tr>
<tr>
<td>1 0 0 0</td>
<td></td>
</tr>
<tr>
<td>0 0 0 1</td>
<td></td>
</tr>
<tr>
<td>0 1 0 0</td>
<td></td>
</tr>
</tbody>
</table>

The assignment problem can also be described mathematically. The goal is to find an assignment to maximize \( Z \) where \( Z = \sum \sum c_{ij} \cdot a_{ij} \) subject to the following constraints:

1. \( \sum_{i=1}^{N} a_{ij} = 1 \) for all \( j \),
2. \( \sum_{j=1}^{N} a_{ij} = 1 \) for all \( i \),
3. \( a_{ij} \in \{0, 1\} \) where \( c_{ij} \) is the rate assistant \( i \) reshelves area \( j \) and \( a_{ij} = 1 \) if worker \( i \) is assigned to task \( j \).

Before presenting some of the existing techniques used to solve the Assignment Problem, a neural network approach, the main focus of this project, will be presented in Chapter IV.
CHAPTER IV
THE ASSIGNMENT PROBLEM AS A NEURAL NETWORK

Introduction

Tank and Hopfield (1987) have suggested that a neural network could be used to solve the assignment problem. The real world examples from Chapter I show that each assignment involves gain/loss for the total problem and this gain/loss equates with the excitatory/inhibitory connections in a neural network. The matrix and graph models of the assignment problem suggest that a neural network could be developed which would solve the assignment problem. The matrix model, with a solution indicated, shows the elements which could have an excitatory or inhibitory effect on a specific processing element. Processing elements in the same row or column would have an inhibitory effect; while all other processing elements would have a excitatory effect. The graph model of the assignment problem suggests a way to express the connectivity between various processing elements.

The Initial Neural Networks

In designing a neural net model for the assignment problem, the question of what each processing element should represent was tackled first. An initial attempt which used input processing elements to represent workers and output processing elements to represent tasks failed to produce results worth further study. Both input and output processing elements henceforth represent task/worker pairs.

The question of connectivity was addressed next. In the initial approach, each output processing element received excitatory inputs from input processing elements which represented different tasks and workers. Each output processing element received inhibitory inputs from input processing elements which represented the same worker or task. Output
processing element $PE_{ij}$ received excitatory inputs from all input processing elements $PE_{k,l}$ such that $k \neq i$ and $l \neq j$. Output processing element $PE_{ij}$ received inhibitory inputs from all input processing elements $PE_{k,l}$ if $k = i$ or $l = j$. The weight of these connections was set at the rate worker $i$ performed task $j$. No connection existed between output processing element $PE_{ij}$ and input processing element $PE_{k,l}$ if $k = i$ and $l = j$. The process of assignment was made by selecting the worker/task pair with the largest net input, eliminating related processing elements and repeating this process until all assignments had been made. The process of selecting the worker/task element with the largest net input is an example of a "winner-take-all" operation. In a winner-take-all operation, the processing element with the largest net input sends an inhibitory signal to the other processing elements which turns them off. Figure 7 shows the neural network described in this section.

This approach failed to generate good assignments for two reasons: (1) the rate at which worker $i$ performed task $j$ was ignored, and (2) the influence of the excitatory inputs was clearly overstated when the number of workers exceeded three. For example, when ten workers are to be assigned to ten tasks, each output processing element received eighteen inhibitory inputs and eighty-one excitatory inputs. To correct these two problems, an excitatory connection was added between input processing element $PE_{k,l}$ and output processing element $PE_{ij}$ and a new propagation rule was developed so activation would occur around zero as opposed to the largest values. The weight from input $PE_{ij}$ to output $PE_{ij}$ was multiplied by $(N - 1)^2$ and weights from input $PE_{k,l}$ to output $PE_{ij}$ were multiplied by $N - 1$. This resulted in the following propagation rule (Processing element 1,1 used for convenience):

$$Net_{ij} = (N-1)^2 \times w_{t_{ij}} + \sum w_{tk} - N-1 \times (\sum w_{t_{ij}} + \sum w_{tk}) \text{ where } k \neq 1 \text{ and } l \neq 1.$$  

The revised neural network is shown in Figure 8.

Results were calculated for all output processing elements in the neural network. The
neural network then used a winner-take-all operation, as in the original model. When an output processing element was assigned, all related output processing elements were turned off and the winner-take-all proceeded among the remaining elements. For example, if processing element 2,5 was chosen for assignment, then all processing elements with worker 2 or task 5 would be turned off and the process would continue from that point. This attempt at a neural network solution produced results that were better than the original neural network, but the results were still clearly unacceptable. The excitatory input from unrelated processing elements tended to force assignments that left the remaining matrix with the greatest potential while not selecting the best processing element for current assignment.

The third attempt at the solution eliminated the excitatory inputs from unrelated processing elements (those with different worker and task). The propagation rule was

Figure 7. Initial Neural Network Representation of the Assignment Problem.
modified to reflect this change. The new rule becomes $\text{Net}_{i,j} = 2 * N * w_{i,j} - \Sigma w_{i,j} - \Sigma w_{k,l}$

where $k \neq 1$ and $l \neq 1$. Assignments were made in the same fashion as in the previous neural networks. This neural network is shown in Figure 9.

Addition of Hidden Layers to the Neural Network

The removal of the excitatory connections between processing elements in differing rows and columns was an improvement; but not a total success because assignments were not optimal. The assignments should not be made on an absolute basis, but an assignment should be made because it is clearly the best choice among related processing elements. To put this idea into effect required the addition of two hidden layers of processing elements. The first hidden layer of $N^2$ processing elements received the same inputs as the old output processing
elements had received, i.e., $\text{Net}_{1,t}$ to hidden layer $= 2 \cdot N \cdot w_{t,1} - \sum w_{t,1} - \sum w_{k,t}$. The second hidden layer consisted of $N$ processing elements with each processing element representing a common worker. These processing elements received input directly from the processing elements in the other hidden layer and used the propagation rule: $\text{Net}_i = \text{second} \{w_{i,1}\}$, i.e., the second largest input to that processing element. The two hidden layers provided inputs to the output processing elements, hidden layer one provided excitatory inputs and hidden layer two provided inhibitory inputs. The assignments were then chosen with a winner-take-all approach and subsequent reduction as previously. The propagation rule for the output processing elements became $\text{Net}_{t,1} = w_{t,1}$ from hidden layer one - $w_i$ from hidden layer two.

This neural network is shown in Figure 10.

This neural network did not take the tasks into consideration, so another hidden layer
of N processing elements was added. Each element in this layer represented a common task. The propagation rule for this layer was \( \text{Net}_i = \text{second} \{w_{i1}\} \), i.e., the second largest input to that processing element. The propagation rule for the output processing elements was rewritten as \( \text{Net}_{i,2} = w_{i,1} \) from hidden layer one - max \( \{w_i\} \) from worker hidden layer or task hidden layer. The resulting neural network is similar to that shown in figure 10 with the addition of a task hidden layer and a change in the output processing element’s propagation rule.

Two additional changes were made to the neural network, the task and worker hidden layers were both allowed to provide an inhibitory signal to the output processing elements and the excitatory input from hidden layer one was doubled (offsetting the increased
inhibitory inputs). This change was made, because both the task and worker elements should have an effect on the chosen assignment, not just the larger of the elements. The final propagation rule for the output processing layers was $Net_{t1} = 2 \times w_{t1}$ from hidden layer one - $w_t$ from task hidden layer - $w_w$ from worker hidden layer. Assignments were still chosen on a winner-take-all basis with subsequent reduction in the model. This reduction was added as a reset line from the output processing elements to related input processing elements. For example, if output processing element $PE_{34}$ is allocated, an inhibitory signal is sent to all input processing elements $PE_{31}$ and $PE_{41}$. The final neural network developed to solve the assignment problem is shown in Figure 11.

![Final Neural Network Model for the Assignment Problem](image)

Figure 11. Final Neural Network Model for the Assignment Problem.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Experiments With the Neural Network Algorithm

Through this point in the development of the neural network results had been generated by hand working with small (5 X 5) matrices. The two principal questions to answer at this time were: (1) "Does this neural network make better assignments than "greedy" (Horowitz & Sahni 1978) methods?" and (2) "Is the assignment made by the neural network optimal?"

Three programs were written in Turbo Pascal (Borland 1987) on an IBM-PC to attempt to answer these questions. The programs also allowed testing on larger matrices (up to 50 X 50).

Greedy Method #1

Greedy algorithm #1 (greedy1.pas in Appendix A) made an assignment if (a) the element had maximum value, and (b) the element was the only element in the row or column that had maximum value. When all assignments of this type were made the number of elements in a row or column with maximum value was incremented and the process continued. When no elements of maximum value remained, the value was decremented and the number searched for was reset to one. After each assignment, the matrix was reduced, i.e., the values of elements in the same row or column were zeroed. The program terminated when N assignments had been made in an N X N matrix. This algorithm is shown in pseudo-code in Figure 12.

Greedy Method #2

The second greedy algorithm (greedy2.pas in Appendix A) makes an assignment to an element if that element has a value larger than any other in its row or column. For example, if element 5,3 (in a 10 X 10 matrix with values ranging from 1 to 10) had a value of 8 and no other element in row 5 exceeded 8, then element 5,3 would be assigned. After an assignment
Matchnumber := 1
Searchvalue := maximum in matrix
Repeat until N assignments have been made
(Scan elements in rows followed by columns)
begin
If matchnumber of elements in a row or column have searchvalue
then
assign an element with searchvalue to the solution
Reduce the matrix by zeroing values in the assigned row/column
else
increment matchnumber
if matchnumber > N
then
decrement searchvalue
reset matchnumber to 1
end

Figure 12. Pseudo-Code for Greedy Method #1.

is made, the matrix is reduced by zeroing elements in the same row or column as the assigned element. This process continues until N assignments have been made. If no row or column has a single largest element, then an assignment is made in a row or column with multiple largest elements (beginning with 2 and incrementing). The pseudo-code for this algorithm is shown in Figure 13.

Repeat until N assignments have been made
(Scan elements row-wise then column-wise)
begin
if an element i,j has maxvalue in row i or column j
then
add element i,j to the assignment
reduce the matrix in row i and column j
else
find multiple elements with highest value in a row or column
assign one of these elements
reduce matrix appropriately
end

Figure 13. Pseudo-Code for Greedy Method #2.
Test Results

Each program made assignments to one hundred (100) 10 X 10 matrices with values from 1 to 10. The results showed that (a) the neural network solution produced better solutions than the greedy methods, and (b) because the greedy methods produced a better result on occasion, the neural network did not produce an optimal solution. Table 4 shows the results of this test (individual test results are shown in Appendix B).

Table 4

<table>
<thead>
<tr>
<th>Method</th>
<th>Average</th>
<th>% of NN</th>
<th>&lt; NN</th>
<th>= NN</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEURAL NETWORK</td>
<td>90.45</td>
<td>100</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>GREEDY ONE</td>
<td>86.22</td>
<td>95.32</td>
<td>89</td>
<td>9</td>
</tr>
<tr>
<td>GREEDY TWO</td>
<td>86.41</td>
<td>95.53</td>
<td>76</td>
<td>11</td>
</tr>
</tbody>
</table>

The results of this series of tests created two more questions: (1) How close to an optimal solution did the neural network provide?, and (2) Could some weighting be changed to produce a result closer to optimal? The test to determine closeness to optimal used a run of fifty (50) 10 X 10 matrices with a value range of 1 to 10. The elements at $i,i$ were assigned the value 10, therefore the resulting matrix had a known maximum of 100. Rows were exchanged so that these maximum elements were not all along the diagonal in the matrix. An additional run of fifty (50) 20 X 20 matrices with values from 1 to 20 was tried to see if the size of the matrix affected this result. The results are shown in Table 5 with complete results in Appendix B.
Table 5
Neural Network Results on Matrices With Known Maximums

<table>
<thead>
<tr>
<th>Matrix Size</th>
<th>Weights</th>
<th>Optimal Solution</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 X 10</td>
<td>1 - 10</td>
<td>94</td>
<td>99.92</td>
</tr>
<tr>
<td>20 X 20</td>
<td>1 - 20</td>
<td>46</td>
<td>99.72</td>
</tr>
</tbody>
</table>

The poorer result as the matrix became larger led to the idea of changing the propagation rule to the first hidden layer. This rule was currently $Net_{ij} = 2 \cdot (N - 1) \cdot w_{ij} - \sum w_{ij} - \sum w_{kj}$ where $k \neq i$ and $l \neq j$. Runs were attempted with the coefficient (the multiplier) in the above formula changed to 3, 5, 10, and 20. The idea of increasing this multiplier was attempted because in a larger matrix a wrong assignment tends to have a greater effect and the larger multiplier has the effect of producing a form of greedy method into the neural net,

Table 6
Results with Various Multipliers on Matrices With Known Solutions (50 Trials)

<table>
<thead>
<tr>
<th>Multiplier</th>
<th>% Correct</th>
<th>% of Maximum</th>
<th>% Correct</th>
<th>% of Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>94</td>
<td>99.92</td>
<td>46</td>
<td>99.72</td>
</tr>
<tr>
<td>3</td>
<td>88</td>
<td>99.84</td>
<td>56</td>
<td>99.80</td>
</tr>
<tr>
<td>5</td>
<td>88</td>
<td>99.84</td>
<td>78</td>
<td>99.90</td>
</tr>
<tr>
<td>10</td>
<td>88</td>
<td>99.84</td>
<td>78</td>
<td>99.93</td>
</tr>
<tr>
<td>20</td>
<td>88</td>
<td>99.84</td>
<td>76</td>
<td>99.93</td>
</tr>
</tbody>
</table>
i.e., elements with the largest weights tend to be selected first. The results are shown in Table 6. The results appeared to show that in larger matrices more influence should be attached to the weight of the processing elements. Subsequent testing on matrices without a known solution produced the results shown in Table 7.

These results showed that the original multiplier was probably correct and the results from the early test were different because of the "cooked" data. For the remainder of the tests, the multiplier used was always 2. The multiplier used in the neural network program is decreased by 2 after each assignment and because of the way the inhibitory inputs are summed, the multiplier used is actually $2^* (N - \# \text{ of assignments made})$.

Table 7

<table>
<thead>
<tr>
<th>Multiplier</th>
<th>Raw Total</th>
<th>Raw Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4550</td>
<td>18852</td>
</tr>
<tr>
<td>3</td>
<td>4537</td>
<td>18839</td>
</tr>
<tr>
<td>5</td>
<td>4535</td>
<td>18832</td>
</tr>
<tr>
<td>10</td>
<td>4530</td>
<td>18807</td>
</tr>
<tr>
<td>20</td>
<td>4530</td>
<td>18806</td>
</tr>
</tbody>
</table>

The Neural Network Algorithm

The neural network program consists of five modules: "initialize," "hidden," "findpair," "reduce," and "print." The initialize module is called at the beginning of each assignment.
problem. This module uses the Turbo Pascal random number function to set initial matrix values. The range of values is varied in later tests to show the effect of differing ranges. The module also creates an assignment matrix and a scratch matrix. The sum of the initial weights in each row and column is calculated and retained for later use. The module initialize has a complexity of \( O(N^2) \), where \( N \) is the number of elements in a row or column.

The module entitled hidden calculates the net input to processing elements in the hidden layer of the neural network. The formula used is 
\[
b[x,y] := 2 \times (\text{size} + 1 - \text{passno}) \times a[x,y] - \text{row}[x] - \text{col}[y].
\]

\( A[x,y] \) is the scratch matrix, whose values are continually being reduced. \( \text{Row}[x] \) and \( \text{col}[y] \) are the inhibitory effects from elements within the same row or column. This module has a complexity of \( O(N^2) \), where \( N \) is the number of elements in a row or column.

The procedure findpair selects the next processing element for assignment. This procedure uses the values calculated in procedure hidden to: (a) find the second largest element in each row and column, and (b) calculate the input to the second hidden layer. The procedure also calculates the winner-take-all result on the output processing elements. When an element is selected for assignment, the findpair procedure calls procedure reduce. Procedure findpair has complexity \( O(N^2) \), where \( N \) is the number of elements in a row or column.

Procedure reduce makes the assignment (selected by procedure findpair) to the allocation matrix and reduces the matrix appropriately. This reduction is the equivalent of sending an inhibitory signal to each input processing element in the same row or column as the processing element just assigned. The complexity of procedure reduce is \( O(N) \), where \( N \) is the number of elements in a row or column.

Procedure print sends the seed number and total value of the assignments to a file entitled "assptest." The procedure also writes the grand total of all problems to the file.
Because procedures hidden, findpair, and reduce are called once for each assignment (N times total), the complexity of the entire program is $O(N^3)$, where $N$ is the number of elements in a row or column. The source code for the neural network program (assprob.pas) is located in Appendix A.
CHAPTER V

OTHER APPROACHES TO THE ASSIGNMENT PROBLEM

Introduction

This section discusses three approaches to the assignment problem: (1) Vogel's Approximation Method or VAM (Ignizio 1982, pp. 291-296), (2) branch-and-bound (Lawler & Wood 1966), and (3) the Hungarian Method. Branch-and-bound and the Hungarian Method yield optimal results directly; although branch-and-bound is not polynomial.

Vogel's Approximation Method (VAM)

VAM utilizes the following steps to produce an approximate solution to the assignment problem. The steps are illustrated in the accompanying tables (a minimization example in Tables 8 - 12).

1. To determine a minimum or maximum solution utilize the matrix model of the assignment problem, assign a penalty value to each row and column (PR, and PC). The penalty is the difference between the largest (or smallest for minimization) weight in a row or column and the second largest (or second smallest for minimization) weight in that row or column. The penalties are shown in the right-hand column and bottom row of the tables.

2. Determine the largest penalty. The largest penalty is shown in bold on the tables.

3. Make an assignment to the largest (or smallest for minimization) element in the row or column found in step 2. The element which has the largest penalty is shown in bold italics on the accompanying tables.

4. Reduce the matrix in the row and column assigned.
5. Repeat steps 1 through 4 until all assignments have been made.

**Table 8**

*Original Matrix With Initial Penalties - $PR_i$ & $PC_j$*

<table>
<thead>
<tr>
<th>Column</th>
<th>Row</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>$PR_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
<td>41</td>
<td>17</td>
<td>15</td>
<td>18</td>
<td>13</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>14</td>
<td>20</td>
<td>17</td>
<td>15</td>
<td>53</td>
<td>15</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>22</td>
<td>23</td>
<td>25</td>
<td>40</td>
<td>27</td>
<td>28</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>18</td>
<td>50</td>
<td>20</td>
<td>19</td>
<td>27</td>
<td>25</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>8</td>
<td>20</td>
<td>19</td>
<td>10</td>
<td>30</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>23</td>
<td>25</td>
<td>25</td>
<td>30</td>
<td>25</td>
<td>43</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

PC$_j$  5   12  0   0   8   2

**Table 9**

*Matrix After First Reduction (Row 5, Column 2) With New Penalties - $PR_i$ & $PC_j$*

<table>
<thead>
<tr>
<th>Column</th>
<th>Row</th>
<th>1</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>$PR_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
<td>17</td>
<td>15</td>
<td>18</td>
<td>13</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>14</td>
<td>17</td>
<td>15</td>
<td>53</td>
<td>15</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>22</td>
<td>25</td>
<td>40</td>
<td>27</td>
<td>28</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>18</td>
<td>20</td>
<td>19</td>
<td>27</td>
<td>25</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>23</td>
<td>25</td>
<td>30</td>
<td>25</td>
<td>43</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

PC$_j$  2   0   0   7   2
Table 10

Matrix After Second Reduction (Row 1, Column 5) With New Penalties - PR, & PCj

<table>
<thead>
<tr>
<th>Column</th>
<th>Row</th>
<th>1</th>
<th>3</th>
<th>4</th>
<th>6</th>
<th>PR,</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>14</td>
<td>17</td>
<td>15</td>
<td>15</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>22</td>
<td>25</td>
<td>40</td>
<td>28</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>18</td>
<td>20</td>
<td>19</td>
<td>25</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>23</td>
<td>25</td>
<td>30</td>
<td>43</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>PCj</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 11

Matrix After Third Reduction (Row 2, Column 6) With New Penalties - PR, & PCj

<table>
<thead>
<tr>
<th>Column</th>
<th>Row</th>
<th>1</th>
<th>3</th>
<th>4</th>
<th>PR,</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>22</td>
<td>25</td>
<td>40</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>18</td>
<td>20</td>
<td>19</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>23</td>
<td>25</td>
<td>30</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>PCj</td>
<td>4</td>
<td>5</td>
<td>11</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Final assignment is made at element 6,3. The complete assignment (1,5), (2,6), (3,1), (4,4), (5,2), and (6,3) with a total weight of 107 units. A second algorithm, Modified Distribution or MODI (Ignizio 1982, pp. 296-301), can be combined with VAM to produce an optimal result. VAM is used to produce the non-optimal feasible solution because the method VAM uses produces a reasonable initial assignment. The MODI algorithm requires the following steps.
Table 12

| Matrix After Fourth Reduction (Row 4, Column 4) With New Penalties - PRᵢ & PCᵢ |
|-------------------|-------------------|-------------------|
| Column            | Row 1            | 3                | PRᵢ           |
|                   | 3                | 22               | 25             | 3              |
|                   | 6                | 23               | 25             | 2              |
| PCᵢ              | 1                | 0                |                |

1. A non-degenerate feasible solution is required. Because the assignment problem solution is degenerate, the solver must add \((N - 1)\) ε assignments (for a total of \(2 \times N - 1\) assignments). An ε assignment is considered to be approaching zero and will be zeroed when an optimal solution is found. The \((N - 1)\) ε assignments must be made to independent empty elements. An element is considered independent, if no Θ path can be constructed beginning at that element. A Θ path starts and ends at the initial element and with only horizontal and vertical moves must trace a path through assigned elements.

2. MODI numbers (\(Rᵢ\) for rows, \(Dᵢⱼ\) for unassigned elements, and \(Cⱼ\) for columns) are added. \(Rᵢ\) and \(Cⱼ\) are given values so that \(Rᵢ + Cⱼ + wᵢⱼ = 0\) in assigned elements (the ε assignments are included as assignments). \(Dᵢⱼ\) is assigned to each unassigned element with the value \(wᵢⱼ + Rᵢ + Cⱼ\). As examples, element 1,1 has received an ε assignment so \(w₁₁ + R₁ + C₁ = 0\) (12 + (-12) + 0) and element 5,3 is unassigned so \(w₅₃ + R₅ + C₃ = D₅₃\) (20 + (-4) + (-3) = 13). The original MODI with initial penalties is shown in Table 13.

3. Assignment is changed beginning at element which has smallest \(Dᵢⱼ\). If no element has \(wᵢⱼ < 0\), then the solution is optimal, else repeat steps 2 and 3.

VAM with MODI does not perform well on the assignment problem because the solution
found by VAM is always degenerate and \((N - 1)\) additional \(\varepsilon\) assignments must be made.

The closer a solution is to optimal, the fewer iterations of MODI are required. VAM has a complexity of \(O(N^3)\) and a program was written in Turbo Pascal (vam.pas found in Appendix A) to compare the results of VAM with the neural network approach. These results will be discussed in the next chapter of this paper.

### Branch-and-Bound Techniques

The assignment problem can be solved using a branch-and-bound technique. Despite having a worst case complexity (Taha 1975, pp. 139-176) of \(O(N!)\), most instances are solvable in polynomial time. Branch-and-bound techniques applied to the assignment problem are

---

<table>
<thead>
<tr>
<th>Row</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>(R_i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(\varepsilon)</td>
<td>25</td>
<td>2</td>
<td>-18</td>
<td>A</td>
<td>-20</td>
<td>-12</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>22</td>
<td>20</td>
<td>(\varepsilon)</td>
<td>53</td>
<td>A</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>A</td>
<td>-3</td>
<td>(\varepsilon)</td>
<td>-3</td>
<td>-1</td>
<td>-15</td>
<td>-22</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>48</td>
<td>19</td>
<td>A</td>
<td>23</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>A</td>
<td>13</td>
<td>-15</td>
<td>(\varepsilon)</td>
<td>5</td>
<td>-4</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>-1</td>
<td>A</td>
<td>-13</td>
<td>-3</td>
<td>(\varepsilon)</td>
<td>-22</td>
</tr>
<tr>
<td>(C_j)</td>
<td>0</td>
<td>-4</td>
<td>-3</td>
<td>-21</td>
<td>-6</td>
<td>-21</td>
<td></td>
</tr>
</tbody>
</table>

\(A\) = Assigned by VAM  
\(\varepsilon\) = Added for Non-Degenerate Solution
shown using the same example that was used to explain VAM.

1. A lower bound is established. The lower bound ($Z_n$) is the sum of the smallest weight in each row. If the elements selected to produce this sum form a feasible (one element only in each row and column) solution, then the selected elements constitute the optimal solution with the bound being the value of the solution. If the solution is not feasible, then branching occurs. In the example problem, the lower bound is ninety-six (96) and the solution is not feasible.

Figure 14. First Step in Branch-and-Bound Technique.

2. Elements $i,1$ are assigned and a new lower bound is determined. These lower bounds are the sum of the value of element $i,1$ and the smallest value (not in column 1) in each remaining row. This technique produces $N$ (6 in the example) branches from the original
node. Each node is evaluated for feasibility. Branching continues from the node with the smallest lower bound until a node with a feasible solution and the smallest lower bound is reached. Each branching from the second level will have \( N - 1 \) nodes and the number of branches per node is reduced by one for each level of the tree.

In the example shown, the first branching results in all non-feasible solutions (Figure 14). The assignment of element 6,1 results in the lowest lower bound; therefore, branches to node in column 2 are made from this node. The lower bounds on these nodes are greater than on the first column of nodes and the algorithm continues from the node with the lowest lower bound (Figure 20). Figures 15, 16, 17, 18, and 19 show the branches from the other first level nodes. The lowest bound in these columns is found in the path that begins with 3,1 and 5,2. On the third level of this path a feasible solution of 103 is found and no other path has a solution which can be lower than this path (figure 21). The solution is (3,1), (5,2), (4,3), (2,4), (6,5), and (1,6) with a total weight of 103. As can be seen in the figures, not all branches must be followed throughout the entire tree.

![Figure 15. Branching From 1,1.](image1)

![Figure 16. Branching From 2,1.](image2)
When a feasible solution is found with a value lower than the lower bound on a non-feasible path, that non-feasible path is not further evaluated. Branch-and-bound is a quasi-enumerative technique that is fairly efficient on small problems and most problems that present obvious choices (begin the branch-and-bound on a column that has one value at max and all others at minimum), but it is still too inefficient for large problems.
Figure 21. Branching From Node 3,2 Which Leads to the Final Assignment.

The Hungarian Method

The assignment problem can be solved in polynomial time with an algorithm developed by H. W. Kuhn (1955). This algorithm is known as the "Hungarian" Method in honor of the Hungarian mathematician Egervary. The Hungarian Method consists of five subroutines: (1) initial reduction, (2) initial assignment, (3) labelling, (4) assignment change, and (5) further reduction.

Since the Hungarian Method solves the example problem shown for the branch-and-
bound and VAM methods using only the first two subroutines, the example problem is shown in Figure 22; while a more complex problem (Figures 23 - 28) is discussed within this chapter.

Figure 22. The Hungarian Method's Solution to the Example Problem.

Initial Reduction

The Hungarian Method uses a matrix representation of the assignment problem and the initial reduction subroutine guarantees that one or more zeroes will be present in each row and column. The steps in this subroutine are as follows:

1. Using the initial matrix, the minimum weight in each row is found (Figure 23).

2. The row minimum is subtracted from each weight in its row to produce the row reduced matrix (Figure 24).
Figure 23. Initial Matrix Showing Row Minimums.
The row minimums are shown in this figure as the large numbers and in the right-hand column. The row minimums are used to calculate the total weight of the final assignment. In this example, \( \Sigma \) Row Minimums = 22.
3. Using the row reduced matrix, the minimum weight in each column is found (Figure 24).

4. The column minimum is subtracted from each weight in its column to produce the fully reduced matrix (Figure 25).

5. The row minimums of step 1 and the column minimums of step 3 are summed and retained for calculation of the final assignment weight. In the example problem, the total weight to retain is 27 as shown in Figure 24.

The initial reduction subroutine of the Hungarian Method has a complexity of $O(N^3)$ where $N$ is the number of elements in a row or column.

Initial Assignment

The initial assignment subroutine makes assignments to elements with a weight of zero in the fully reduced matrix until $N$ assignments have been made or no zeroes remain. The subroutine marks off a row and a column after each assignment to prevent further assignments in that row or column. The steps in this subroutine are as follows:

1. Using the fully reduced matrix, the weights are scanned row-by-row and column-by-column searching for a row or column that contains a single zero.

2. When such a row or column is found, the element with a weight of zero is assigned.

3. The elements in the assigned row or column are marked as assigned and their weights are no longer used in evaluating a row or column for a single zero. For example, if element 3,4 is assigned, weights in row 3 and column 4 are no longer considered.

4. This procedure continues until one of three conditions occur: (1) $N$ assignments have been made, (2) fewer than $N$ assignments have been made and no weights of zero remain, or (3) fewer than $N$ assignments have been made and all unassigned rows and columns have more than one weight of zero. If condition 1 occurs, the assignment is a minimum.
<table>
<thead>
<tr>
<th>ROW #</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7</td>
<td>4</td>
<td>9</td>
<td>0</td>
<td>12</td>
<td>1</td>
<td>12</td>
<td>4</td>
<td>8</td>
<td>9</td>
<td>6</td>
<td>5</td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>7</td>
<td>1</td>
<td>11</td>
<td>2</td>
<td>11</td>
<td>6</td>
<td>2</td>
<td>9</td>
<td>0</td>
<td>6</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>11</td>
<td>2</td>
<td>4</td>
<td>10</td>
<td>0</td>
<td>7</td>
<td>9</td>
<td>0</td>
<td>2</td>
<td>12</td>
<td>9</td>
<td>11</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>1</td>
<td>12</td>
<td>8</td>
<td>7</td>
<td>1</td>
<td>2</td>
<td>9</td>
<td>2</td>
<td>9</td>
<td>1</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>12</td>
<td>10</td>
<td>11</td>
<td>11</td>
<td>9</td>
<td>5</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>8</td>
<td>12</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>9</td>
<td>1</td>
<td>2</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>3</td>
<td>8</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>7</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>9</td>
<td>10</td>
<td>5</td>
<td>9</td>
<td>3</td>
<td>4</td>
<td>11</td>
<td>6</td>
<td>0</td>
<td>2</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>8</td>
<td>4</td>
<td>6</td>
<td>7</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>7</td>
<td>10</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>3</td>
<td>11</td>
<td>4</td>
<td>10</td>
<td>5</td>
<td>6</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>11</td>
<td>8</td>
<td>6</td>
<td>7</td>
<td>3</td>
<td>6</td>
<td>4</td>
<td>9</td>
<td>2</td>
<td>11</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>11</td>
<td>7</td>
<td>12</td>
<td>0</td>
<td>7</td>
<td>2</td>
<td>9</td>
<td>9</td>
<td>10</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>10</td>
<td>0</td>
<td>7</td>
<td>7</td>
<td>0</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>COL MIN</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 24.** Row Reduced Matrix Showing Column Minimums.
The larger characters in the figure represent the column minimums. The column minimums are used to calculate the weight in the final assignment. In this example, \( \Sigma \) column minimums = 5. The sum of the column minimums and the row minimums is 27.
assignment with weight equal to the sum found in step 5 of the initial reduction subroutine and the Hungarian Method terminates. If condition 2 occurs, the Hungarian Method moves on to the labelling subroutine. If condition 3 occurs, an assignment is made arbitrarily to an element with a weight of zero in the row or column with the fewest number of zeroes. Note: Marking off the elements in the row and column of this assignment may result in rows or columns which now have a single zero; the zero element in these rows or columns should be assigned before further arbitrary assignments are made.

The example problem with the 6 X 6 matrix is shown in Figure 22 and terminates after this subroutine with an optimal assignment. Figure 25 illustrates the initial assignment subroutine on the more complex problem. The initial assignment subroutine has a complexity of O(N^3).

Labelling

The labelling subroutine determines if an additional assignment can be made to the current matrix or if the current matrix must be reduced further to make additional assignments. If an additional assignment can be made, the breakthrough subroutine is called. If the matrix must be reduced further, the further reduction subroutine is called. The steps in the labelling routine are as follows:

1. All old labels are removed.

2. Using the current reduced matrix (initially the fully reduced matrix from the initial reduction subroutine) and the current assignments (initially the assignments made in the initial assignment subroutine), label all rows that do not have an assignment with a "+.

3. Find all elements in a labelled row that would be an acceptable assignment, i.e., all elements with weight equal to zero. Label these columns with the row number of the row being examined. If a column already has a label, do not change the label.
<table>
<thead>
<tr>
<th>ROW #</th>
<th>COLUMN #</th>
<th>ASGN ORDER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 2 3 4 5 6 7 8 9 10 11 12 13</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>7 3 9 0 12 1 10 4 7 9 6 4 11</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>10 6 1 11 2 1 9 6 1 9 0 5 10</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>11 1 4 10 0 7 7 0 1 12 9 10 5</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>8 0 12 8 7 1 0 9 1 9 1 0 0</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>1 4 1 2 3 12 8 11 10 9 5 2 0</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>0 7 12 3 4 6 4 0 8 1 2 6 6</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>10 2 8 3 0 0 2 7 9 9 8 1 3</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>4 8 10 5 9 3 2 11 5 0 2 6 5</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>6 1 0 4 8 4 4 7 0 2 1 5 1</td>
<td>7</td>
</tr>
<tr>
<td>10</td>
<td>7 9 1 5 0 3 9 4 9 5 6 0 6</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>8 5 7 3 6 4 7 2 1 6 0 7 7</td>
<td>N/A</td>
</tr>
<tr>
<td>12</td>
<td>11 6 12 0 7 2 7 9 9 4 2 2 1</td>
<td>N/A</td>
</tr>
<tr>
<td>13</td>
<td>5 1 5 5 2 2 8 0 6 7 0 2 2</td>
<td>9</td>
</tr>
</tbody>
</table>

Figure 25. Fully Reduced Matrix With Initial Assignments. The larger characters in the figure are those that have been added to the assignment. All the assignments are only made to elements with a value of zero. The order of the assignments is shown in the right-hand column. Assignments 1 - 5 were made because the element chosen was the only remaining element in its row with a zero value. Assignments 6 - 10 were made because the element chosen was the only remaining element in its column with a value of zero. Assignment 11 was made because its element was the only element in the matrix which still had a value of zero.
4. Find the assigned element in each labelled column and label the row this assignment is in with the number of the column being examined.

5. The subroutine terminates when one of two conditions occur: (1) a column without an assignment is labelled or (2) no more labels can be added. If condition 1 occurs, another assignment can be added to the matrix and the program jumps to the breakthrough subroutine. If condition 2 occurs, the matrix must be reduced further before another assignment can be made and the program jumps to the further reduction subroutine.

Labelling resulting in jumps to the further reduction subroutine are shown in Figures 26 and 27. Labelling resulting in a jump to the breakthrough subroutine is shown in Figure 28. The labelling subroutine has a complexity of $O(N^2)$.

**Further Reduction**

This subroutine reduces the weights in certain elements of the current matrix to ensure another assignment is possible. The subroutine does not change assignments and upon completion of the reduction, the program returns to the labelling subroutine. The steps in the further reduction subroutine are as follows:

1. Using the current matrix with the current labelling, find the minimum weight among all elements in labelled rows and unlabelled columns.

2. Subtract the value found in step 1 from all elements in labelled rows and unlabelled columns.

3. Add the value found in step 1 to all elements in unlabelled rows and labelled columns.

4. Find the difference between the number of labelled rows and labelled columns. In the example of Figure 26, there are 4 labelled rows and 2 labelled columns.

5. Multiply the value found in step 1 by the value found in step 4 and add the result to the sum found in step 5 of the initial reduction subroutine. In the example of Figure 26, the
Figure 26. Fully Reduced Matrix With Initial Assignments and Labelling.
Larger characters represent the initial assignments. The row labels (RLBL) are shown in the right-hand column and the column labels (CLBL) are shown in the bottom row. The labels are used to determine whether an additional assignment can be made (breakthrough) or if the matrix requires a further reduction. In this example, no breakthrough occurs (no unassigned column is labelled). The shaded area (labelled rows, unlabelled columns) will be reduced by the smallest value in this area (1). The italicized elements (labelled columns, unlabelled rows) will be incremented by the same amount as the reduced area is reduced (1). The reduction adds to the weight of the final assignment. This amount is number of labelled rows (4) minus number of labelled columns (2) times the reducing value (1). This amount is added to the previous total and is now 29.

<table>
<thead>
<tr>
<th>ROW #</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>RLBL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7</td>
<td>3</td>
<td>9</td>
<td>0</td>
<td>12</td>
<td>1</td>
<td>10</td>
<td>4</td>
<td>7</td>
<td>9</td>
<td>6</td>
<td>4</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>6</td>
<td>1</td>
<td>11</td>
<td>2</td>
<td>1</td>
<td>9</td>
<td>6</td>
<td>1</td>
<td>9</td>
<td>0</td>
<td>5</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>11</td>
<td>1</td>
<td>4</td>
<td>10</td>
<td>0</td>
<td>7</td>
<td>7</td>
<td>0</td>
<td>1</td>
<td>12</td>
<td>9</td>
<td>10</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>0</td>
<td>12</td>
<td>8</td>
<td>7</td>
<td>1</td>
<td>0</td>
<td>9</td>
<td>1</td>
<td>9</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>12</td>
<td>8</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>7</td>
<td>12</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td>4</td>
<td>9</td>
<td>8</td>
<td>1</td>
<td>2</td>
<td>6</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>2</td>
<td>8</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>7</td>
<td>9</td>
<td>8</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>8</td>
<td>10</td>
<td>5</td>
<td>9</td>
<td>3</td>
<td>2</td>
<td>11</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>6</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>8</td>
<td>4</td>
<td>7</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>7</td>
<td>9</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>3</td>
<td>9</td>
<td>4</td>
<td>9</td>
<td>5</td>
<td>6</td>
<td>0</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>8</td>
<td>5</td>
<td>7</td>
<td>3</td>
<td>6</td>
<td>4</td>
<td>7</td>
<td>2</td>
<td>1</td>
<td>6</td>
<td>0</td>
<td>7</td>
<td>7</td>
<td>+</td>
</tr>
<tr>
<td>12</td>
<td>11</td>
<td>6</td>
<td>12</td>
<td>0</td>
<td>7</td>
<td>2</td>
<td>7</td>
<td>9</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>+</td>
</tr>
<tr>
<td>13</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>8</td>
<td>0</td>
<td>6</td>
<td>7</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
minimum weight is 1, the row/column difference is 2, and the previous minimum from the initial reduction subroutine is 27, so the new minimum possible solution has a value of 29. This value is the new minimum possible value for a feasible solution.

The elements which will be decreased are shown on Figures 26 and 27 within the shaded areas. The elements which will be increased are shown on Figures 26 and 27 in italics. Each reduction guarantees that the next pass through the labelling subroutine will find at least one more assignment. The further reduction subroutine has a complexity of \( O(N^2) \).

**Breakthrough**

The breakthrough subroutine is called when the labelling subroutine has determined that an additional assignment can be made to the current matrix. This subroutine will increase the number of assignments by 1 and may change some of the current assignments. The steps in the breakthrough subroutine are as follows:

1. Let column equal the number of the column which caused the breakthrough (i.e. the unassigned column which became labelled). Let row equal the column label of this column. In the example shown in Figure 28, column equals 7 and row equals 4.

2. Make an assignment at element row, column (Element 4,7 in the example).

3. At this point, one of two conditions will occur: (1) the row label of the element assigned in step 2 is a number or (2) the row label of the element assigned in step 2 is "+.

4. If condition 1 has occurred, then let column equal row label and remove the assignment at element row,column. Let row equal the column label of this new column and return to step 2. In the example shown in Figure 28, column is reassigned the value 2, the assignment at element 4,2 is removed, and row is reassigned the value 3.

5. If condition (2) of step 3 has occurred, then the breakthrough subroutine is complete.

The program returns to the labelling subroutine if fewer than N assignments have been
When the example is first labelled, the trivial assignment at element 11,9 is found. This assignment is made and labelling begins anew. Row labels (RLBL) are shown in the right-hand column; column labels (CLBL) are shown in the bottom row. The large zeroes represent the current assignment. The labelling reveals that no additional assignment is possible. The elements within the shaded areas (labelled rows, unlabelled columns) are those which must be reduced. The amount these elements are reduced by is the smallest value within the shaded areas (1). The elements represented by italicized characters (labelled columns, unlabelled rows) will be incremented by the same value. The final weight of the assignment is increased by the number of rows labelled (6) minus the number of columns labelled (5) times the reducing value (1). The new minimum assignment has a weight of 30.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 28. Matrix After Second Reduction With New Labelling and Final Assignment.
The row labels (RLBL) are shown in the right-hand column; the column labels (CLBL) are shown in the bottom row. An additional assignment can be made when column 7, an unassigned column, receives a label (shown shaded). The breakthrough technique involves making and removing assignments in sequence until a row with label of "+" is reached. In this example, the process begins by making an assignment at (7,4) and removing the assignment at (4,2). The row labels and column labels indicate where the next addition and removal is to occur. The addition and removal continues with add at (3,2), remove at (3,5), add at (7,5), remove at (7,6), add at (1,6), remove at (1,4), and add at (12,4). The label on row 12 is "+" and the process terminates. The final assignment is shown in larger characters, while the removed assignments are shown shaded. The weight of the final assignment is 30.

<table>
<thead>
<tr>
<th>ROW #</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>1</td>
<td>7</td>
<td>0</td>
<td>11</td>
<td>0</td>
<td>8</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>5</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>5</td>
<td>0</td>
<td>12</td>
<td>2</td>
<td>1</td>
<td>8</td>
<td>6</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>4</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>0</td>
<td>3</td>
<td>11</td>
<td>0</td>
<td>7</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>11</td>
<td>9</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>0</td>
<td>12</td>
<td>10</td>
<td>8</td>
<td>2</td>
<td>0</td>
<td>10</td>
<td>1</td>
<td>9</td>
<td>2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>12</td>
<td>7</td>
<td>11</td>
<td>9</td>
<td>8</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>13</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>7</td>
<td>12</td>
<td>5</td>
<td>5</td>
<td>7</td>
<td>4</td>
<td>1</td>
<td>8</td>
<td>1</td>
<td>3</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>1</td>
<td>7</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>8</td>
<td>10</td>
<td>7</td>
<td>10</td>
<td>4</td>
<td>2</td>
<td>12</td>
<td>5</td>
<td>0</td>
<td>3</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>1</td>
<td>0</td>
<td>6</td>
<td>9</td>
<td>5</td>
<td>4</td>
<td>8</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>7</td>
<td>9</td>
<td>1</td>
<td>7</td>
<td>1</td>
<td>4</td>
<td>9</td>
<td>5</td>
<td>9</td>
<td>5</td>
<td>7</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>11</td>
<td>7</td>
<td>4</td>
<td>6</td>
<td>4</td>
<td>6</td>
<td>4</td>
<td>6</td>
<td>2</td>
<td>0</td>
<td>5</td>
<td>2</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>12</td>
<td>9</td>
<td>4</td>
<td>10</td>
<td>0</td>
<td>6</td>
<td>1</td>
<td>5</td>
<td>8</td>
<td>7</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>+</td>
</tr>
<tr>
<td>13</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>6</td>
<td>2</td>
<td>2</td>
<td>7</td>
<td>0</td>
<td>5</td>
<td>6</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

CLBL: 5 3 5 12 7 1 4 3 3 12 12

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
made. If N assignments have been made, the final assignment is minimum. Total weight of this assignment is either the value found in step 5 of the initial reduction subroutine (if the further reduction subroutine has never been called) or in step 5 of the further reduction subroutine. In the example, the further reduction subroutine is called twice and the value of the minimum assignment is 30.

The breakthrough subroutine is called (N - the number of assignments made by the initial assignment subroutine) times. In the example, the breakthrough subroutine is called twice. The complexity of the breakthrough subroutine is $O(N)$.

Additional Comments

The complexity for the entire Hungarian Method is $O(N^3)$, as the initial assignment subroutine has complexity of $O(N^3)$ and the labelling subroutine with complexity of $O(N^2)$ may be called N-1 times in a worst case. The Hungarian Method can be used to find a maximum assignment (instead of the minimum described) by multiplying all weights by -1 before following the steps outlined above and multiplying the initial value (found in step 5 of the initial reduction subroutine) by -1. A program to find maximum assignments using the Hungarian Method was developed (hungary.pas in Appendix A) to evaluate the results of the neural network solution. The results of this evaluation are found in chapter 6.
CHAPTER VI

TEST RESULTS

Introduction

The neural network program was tested by comparing the assignments it made with the assignments made by a program which used the Hungarian Method and a program which used VAM. Programs were written in Turbo Pascal that performed VAM and the Hungarian Method. A series of tests were performed with each program making assignments on the same initial matrices. Tables 14 through 25 and the accompanying text will describe and summarize each test. The complete test results can be found in Appendix B.

Test # 1 - Basic Performance

The three programs made assignments to fifty (50) 13 X 13 matrices with weights ranging from 1 to 13. The test was run three times using the random number seeds 1 to 50, 51 to 100, and 101 to 150 respectively. The results are summarized in Tables 14 and 15. This test shows that the neural network approach outperformed VAM, but fell short of the Hungarian

Table 14

<table>
<thead>
<tr>
<th>Trial Number</th>
<th>Method</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>Totals</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hungarian</td>
<td>7801</td>
<td>7814</td>
<td>7812</td>
<td>23427</td>
</tr>
<tr>
<td></td>
<td>Neural</td>
<td>7733</td>
<td>7767</td>
<td>7784</td>
<td>23284</td>
</tr>
<tr>
<td></td>
<td>VAM</td>
<td>7699</td>
<td>7712</td>
<td>7733</td>
<td>23144</td>
</tr>
</tbody>
</table>

54
Table 15
Percentage Comparison to the Hungarian Method

<table>
<thead>
<tr>
<th>Trial #</th>
<th>Hungarian</th>
<th>Neural Net</th>
<th>VAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>99.13</td>
<td>98.69</td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>99.40</td>
<td>98.69</td>
</tr>
<tr>
<td>3</td>
<td>100</td>
<td>99.64</td>
<td>98.99</td>
</tr>
<tr>
<td>Total</td>
<td>100</td>
<td>99.39</td>
<td>98.79</td>
</tr>
</tbody>
</table>

The performance of these three algorithms was easily better than the performance achieved by either of the greedy methods tested earlier.

Tests # 2 and # 3 - Changes in Matrix Weights

The second test run on the three algorithms used fifty (50) 13 X 13 matrices with weights ranging from 1 to 26. Three runs were conducted using the seeds 151 - 200, 201 - 250, and 251 - 300 respectively. The third test run on the three algorithms used fifty (50) 13 X 13 matrices with weights ranging from 1 to 7. Two runs were conducted using the random number seeds 301 - 350 and 351 - 400. These tests were conducted to see if the range of weights had any influence on the performance of the algorithms. The summary of results is shown in Tables 16 through 21 and complete results can be found in Appendix B.

The neural network algorithm outperformed VAM on all trials in this series of tests. The performance of the neural network algorithm improved when compared with the Hungarian Method on a percentage basis. The percentage of correct assignments increased as the ranges were widened (a relatively large weight is much more likely to be in the optimal assignment) and narrowed (multiple optimal assignments are more probable).
Table 16
Test Results 13 X 13 Matrix With Weights 1 - 26

<table>
<thead>
<tr>
<th>Trial Number</th>
<th>Method</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>Totals</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hungarian</td>
<td>15296</td>
<td>15401</td>
<td>15360</td>
<td>46057</td>
</tr>
<tr>
<td></td>
<td>Neural Net</td>
<td>15192</td>
<td>15324</td>
<td>15303</td>
<td>45819</td>
</tr>
<tr>
<td></td>
<td>VAM</td>
<td>15064</td>
<td>15225</td>
<td>15197</td>
<td>45486</td>
</tr>
</tbody>
</table>

Table 17
Percentage Comparison to the Hungarian Method

<table>
<thead>
<tr>
<th>Trial #</th>
<th>Hungarian</th>
<th>Neural Net</th>
<th>VAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>100</td>
<td>99.32</td>
<td>98.48</td>
</tr>
<tr>
<td>5</td>
<td>100</td>
<td>99.50</td>
<td>98.86</td>
</tr>
<tr>
<td>6</td>
<td>100</td>
<td>99.63</td>
<td>98.94</td>
</tr>
<tr>
<td>Total</td>
<td>100</td>
<td>99.48</td>
<td>98.76</td>
</tr>
</tbody>
</table>

Table 18
Number of Correct Solutions Found Compared With Hungarian Method

<table>
<thead>
<tr>
<th>Trial #</th>
<th>Neural Net</th>
<th>VAM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Result =</td>
<td>Result &lt;</td>
</tr>
<tr>
<td>4</td>
<td>22</td>
<td>28</td>
</tr>
<tr>
<td>5</td>
<td>31</td>
<td>19</td>
</tr>
<tr>
<td>6</td>
<td>26</td>
<td>24</td>
</tr>
<tr>
<td>Total</td>
<td>79</td>
<td>71</td>
</tr>
</tbody>
</table>
Table 19

Test Results 13 X 13 Matrix With Weights 1 - 7

<table>
<thead>
<tr>
<th>Trial Number</th>
<th>Hungarian</th>
<th>Neural Net</th>
<th>VAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>4143</td>
<td>4124</td>
<td>4085</td>
</tr>
<tr>
<td>8</td>
<td>4145</td>
<td>4130</td>
<td>4107</td>
</tr>
<tr>
<td>Total</td>
<td>8288</td>
<td>8254</td>
<td>8192</td>
</tr>
</tbody>
</table>

Table 20

Percentage Comparison to the Hungarian Method

<table>
<thead>
<tr>
<th>Trial #</th>
<th>Hungarian</th>
<th>Neural Net</th>
<th>VAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>100</td>
<td>99.54</td>
<td>98.60</td>
</tr>
<tr>
<td>8</td>
<td>100</td>
<td>99.64</td>
<td>99.08</td>
</tr>
<tr>
<td>Total</td>
<td>100</td>
<td>99.59</td>
<td>98.84</td>
</tr>
</tbody>
</table>

Table 21

Number of Correct Solutions Found Compared With Hungarian Method

<table>
<thead>
<tr>
<th>Trial Number</th>
<th>Neural Net Result =</th>
<th>Neural Net Result &lt;</th>
<th>VAM Result =</th>
<th>VAM Result &lt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>36</td>
<td>14</td>
<td>19</td>
<td>31</td>
</tr>
<tr>
<td>8</td>
<td>36</td>
<td>14</td>
<td>27</td>
<td>23</td>
</tr>
<tr>
<td>Total</td>
<td>72</td>
<td>28</td>
<td>46</td>
<td>54</td>
</tr>
</tbody>
</table>

Tests # 4 and # 5 - Matrix Size Changes

Test # 4 increased the size of the matrix to 20 X 20 with weights ranging from 1 to 20.
Two runs of fifty matrices with random number seeds of 401 - 450 and 451 - 500 comprised this test. Test # 5 decreased the size of the matrix to 8 X 8 with weights ranging from 1 to 8 being used. A single run of one hundred matrices with random number seeds of 501 - 600 comprised this test. The results of these tests are given in Tables 22 through 24. Complete results can be found in Appendix B.

The neural network approach outperformed VAM in all cases, but VAM had its best performance in the larger and smaller matrices. Interpretation of the test results will be expanded on in the final section of this chapter and the final chapter will use information gathered from the tests to suggest enhancements.

Table 22
Test Results 20 X 20 Matrix With Weights 1 - 20

<table>
<thead>
<tr>
<th>Trial Number</th>
<th>Method</th>
<th>9</th>
<th>10</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hungarian</td>
<td>19013</td>
<td>19029</td>
<td>38042</td>
</tr>
<tr>
<td></td>
<td>Neural Net</td>
<td>18876</td>
<td>18891</td>
<td>37767</td>
</tr>
<tr>
<td></td>
<td>VAM</td>
<td>18817</td>
<td>18848</td>
<td>37665</td>
</tr>
</tbody>
</table>

Table 23
Percentage Comparison to the Hungarian Method

<table>
<thead>
<tr>
<th>Trial #</th>
<th>Hungarian</th>
<th>Neural Net</th>
<th>VAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>100</td>
<td>99.28</td>
<td>98.97</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td>99.27</td>
<td>99.05</td>
</tr>
<tr>
<td>Total</td>
<td>100</td>
<td>99.28</td>
<td>99.01</td>
</tr>
</tbody>
</table>
Table 24
Test Results 8 X 8 Matrix With Weights 1 - 8

<table>
<thead>
<tr>
<th>Method</th>
<th>Hungarian</th>
<th>Neural Net</th>
<th>VAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total (100 Trials)</td>
<td>5735</td>
<td>5705</td>
<td>5688</td>
</tr>
<tr>
<td>% of Hungarian</td>
<td>100</td>
<td>99.48</td>
<td>99.18</td>
</tr>
</tbody>
</table>

Interpretation of Test Results

The performance of the neural network algorithm varied somewhat depending on the size of the problem and the size of the weights. The algorithm generated its best results (based on percentage of the Hungarian Method) on smaller matrices (8 X 8) and matrices where the range of weights was substantially wider or narrower than the size of the matrix (13 X 13 matrix with weights 1 - 26 and 13 X 13 matrix with weights 1 - 7). The result with wider ranges can be explained by the somewhat greedy nature of the algorithm; a large weight is more likely to be included in the optimal solution, especially when the weight is present less often (on average 1/N weights of 13 and only 1/2N weights of 26). The better result with a narrower range of weights is due to the presence of multiple optimal solutions.

The win/loss record versus the Hungarian Method is best on the matrices with a narrow range of weights (13 X 13 matrix with weights 1 - 7). In all cases the neural network algorithm outperformed the VAM algorithm. Changes in weights did not seem to affect the neural network advantage over VAM, but changes in size give the appearance that VAM was giving an improved result on both larger and smaller matrices although the small sample size may be affecting this result. The first fifty matrices were also run using the two "greedy" methods described previously; methods used on these final tests performed substantially
better.

The neural network approach appears to perform poorer than the Hungarian Method for these reasons: (a) the Hungarian Method changes its assignments throughout the entire process while the neural network and VAM make an assignment and stick with it, and (b) the neural network and VAM can both make an assignment which eliminates an excessive number of good future choices (VAM does this more often, which is why the neural network approach outperforms it).

Although no tests were conducted on the time required to complete a test run, the order of the algorithms from fastest to slowest was (a) Hungarian Method, (b) VAM, and (c) neural network. The final chapter will use these observations and the results of the various tests to suggest possible enhancements and changes which speed up and improve the performance of the neural network algorithm.
CHAPTER VII

CONCLUSION

Miscellaneous Comments

The neural net algorithm could be used as a front-end for the MODI algorithm. The results of the neural net are closer to optimal and using it instead of VAM should reduce the number of assignments required. Since the Hungarian Method solves the assignment problem in polynomial time, this change is probably inadvisable.

The neural net can be used to solve assignment problems where the number of workers and tasks differ. If the workers exceed the tasks, then tasks are added until the quantities are equal. The weights for these added tasks should be larger (for a minimum solution) or smaller (for a maximum solution) than any other weight in the problem. Assignments are made until all workers have been assigned. If tasks exceed workers, then assignments are made until each worker has been assigned one task. All workers are added back into the problem with the weights for unassigned tasks. The process continues until all assignments have been made either by having workers equal tasks or workers exceed tasks when the process described above is followed.

Similarities and Differences

VAM and the neural net are similar in the following respects: (a) both make an assignment and reduce the available choices, (b) neither changes assignments, and (c) both have a winner-take-all method for assignment selection. VAM and the neural net approach differ in that the neural net attempts to develop a total view of the problem while VAM looks at single rows and columns. The major ways in which the Hungarian Method differs from
the other two methods are: (a) the Hungarian Method does not make its assignments from the original matrix, it uses a normalized (reduced) matrix, and the Hungarian Method may make fewer assignments initially; and (b) the Hungarian Method can change the assignments it has made.

Enhancements

The neural net could increase its speed in two ways:

1. Take advantage of the inherent parallelism of a neural net. In this neural net, $N^2$ processors would be required for a fully parallel operation, but speed-up could occur with fewer processors.

2. The neural net makes one assignment per pass. This could be increased by changing the winner-take-all operation to a threshold feature, where all output processing elements with net above the threshold would be turned on.

The performance of the neural net could possibly be improved by using a normalized matrix as a starting point. In the assignment problem, normalization would be accomplished by reducing the matrix in the same manner as in the initial reduction routine of the Hungarian Method. This normalization should reduce the effect on the assignments that occurs because of multiple large or small weights in a single row or column.
Appendix A

Source Code
Source Code for Neural Net Program

program main(input,output);

const
  size = 13;
  trials = 50;
  mult = 2;

var
  start, pass, total, gtotal, seed : integer;
  x, y, z, row1, col1 : integer;
  a, b, c, d : ray;
  col, row : array[1..size] of integer;
  f : text;

procedure initialize;
var
  x, y : integer;
begin (* procedure initialize *)
  for x := 1 to size do
    row[x] := 0;
    col[x] := 0;
  end;
  for x := 1 to size do
    for y := 1 to size do
      a[x,y] := 0;
  randseed := seed;
  for x := 1 to size do
    for y := 1 to size do
      begin
        a[x,y] := random(size) + 1;
        c[x,y] := 0;
        d[x,y] := a[x,y];
        col[y] := col[y] + a[x,y];
        row[x] := row[x] + a[x,y]
      end;
end; (* procedure initialize *)

procedure reduction;
var
  x, y : integer;
begin (* procedure reduction *)
  for x := 1 to size do
    for y := 1 to size do
      begin
        a[x,y] := random(size) + 1;
        c[x,y] := 0;
        d[x,y] := a[x,y];
        col[y] := col[y] + a[x,y];
        row[x] := row[x] + a[x,y]
      end;
end; (* procedure reduction *)
\[ b[x,y] := a[x,y] \times \text{mult} \times \text{(size} + 1 - \text{pass}) - \text{row}[x] - \text{col}[y] \]

else

\[ b[x,y] := 0 \]

end; (* procedure reduction *)

procedure last2;

var

\text{x,y,t : integer;}
\text{m,n : array[1..2]of integer;}

begin (* procedure last2 *)

for t := 1 to 2 do

begin

m[t] := 0;

n[t] := 0

end;

t := 1;

y := 1;

for x := 1 to size do

begin

if row[x] <> 0 then

begin

m[t] := x;

t := t + 1

end;

if col[x] <> 0 then

begin

n[y] := x;

y := y + 1

end

end;

if (a[m[1],n[1]] + a[m[2],n[2]]) >= (a[m[1],n[2]] + a[m[2],n[1]]) then

begin

\text{c[m[1],n[1]] := 1;}

\text{c[m[2],n[2]] := 1}

end

else

begin

\text{c[m[2],n[1]] := 1;}

\text{c[m[1],n[2]] := 1}

end;

end; (* procedure last2 *)

procedure reduce(l,m:integer);
var
n : integer;

begin (* procedure reduce *)
total := 0;
for n := 1 to size do
begin
row[n] := row[n] - a[n,m];
a[n,m] := 0;
col[n] := col[n] - a[l,n];
a[l,n] := 0
end;
row[l] := 0;
col[m] := 0;
c[l,m] := 1
end; (* procedure reduce *)

procedure findpair(var r,c:integer);
var
x,y : integer;
rowmax,colmax,colsec,rowsec,coldiff,rowdiff : array[1..size]of integer;
maxdiff,rindex,cindex : integer;

begin (* procedure findpair *)
for x := 1 to size do
begin
rowmax[x] := 0;
colmax[x] := 0;
rowsec[x] := 0;
colsec[x] := 0
end;
for x := 1 to size do
for y := 1 to size do
begin
if b[x,y] > coldiff
then
coldiff := b[x,y];
if b[x,y] > rowmax[x]
then
rowmax[x] := b[x,y]
end;
for x := 1 to size do
for y := 1 to size do
begin
if b[x,y] <> rowmax[x]
then
if b[x,y] > rowsec[x]
then
rowsec[x] := b[x,y];
if b[x,y] <> colmax[y]
then
  if b[x,y] > colsec[y]
  then
    colsec[y] := b[x,y]
end;
for x:= 1 to size do
begin
coldiff[x] := colmax[x] - colsec[x];
rowdiff[x] := rowmax[x] - rowsec[x]
end;
maxdiff := 0;
ri := 0;
ci := 0;
for x := 1 to size do
  for y := 1 to size do
    if b[x,y] > 0
    then
      if (2* b[x,y] + rowdiff[x] + coldiff[y] - rowmax[x] -
colmax[y]) > maxdiff
      then
        begin
          maxdiff := 2* b[x,y] + rowdiff[x] + coldiff[y] -
rowmax[x] - colmax[y];
          ri := x;
          ci := y
        end;
r := ri;
c := ci;
end; (* procedure findpair *)

procedure print;
var
  x,y : integer;
begin (* procedure print *)
total := 0;
for x := 1 to size do
  for y := 1 to size do
    total := total + d[x,y] * c[x,y];
gtotal := gtotal + total;
write('The seed is ',seed);
writeln(' The total is ',total:5)
end; (* procedure print *)

begin (* program main *)
assign('assptest');
rewrite('f');
writeln('enter starting seed');
readln(start);
writeln(f,'neural net approach unknown maximum');
writeln(f,'the size is ',size:3,' the multiplier is ',mult:3,' the range is ',size:3);
gtotal := 0;
for seed := start to (trials + start - 1) do
begin
initialize;
for pass := 1 to (size - 2) do
begin
reduction;
findpair(row1, col1);
reduce(row1, col1)
end;
last2;
print
end;
writeln(f,'The grand total for all trials is ',gtotal:6);
close(f)
end. (* program main *)

Source Code for Hungarian Method

program main(input, output);

const
n = 13;
trials = 50;
type
r1 = array[1..n] of integer;
r2 = array[1..n,1..n] of integer;
rb = array[1..n] of boolean;
var
a, b, c : r2;
ralloc, calloc : rb;
start, alloc, seed, gtotal : integer;
done : boolean;
f : text;

procedure initialize;
var
x, y : integer;
begin (* procedure initialize *)
randseed := seed;
for x := 1 to n do
for y := 1 to n do
begin
a[x, y] := n - random(n);
b[x, y] := a[x, y]
end
procedure reduction;
var
  x,y : integer;
  rmin,cmin : r1;
begin (* procedure reduction *)
  for x := 1 to n do
    begin
      rmin[x] := maxint;
      cmin[x] := maxint
      end;
  for x := 1 to n do
    for y := 1 to n do
      if b[x,y] < rmin[x]
      then
        rmin[x] := b[x,y];
  for y := 1 to n do
    for x := 1 to n do
      b[x,y] := b[x,y] - rmin[x];
  for y := 1 to n do
    for x := 1 to n do
      if b[x,y] < cmin[y]
      then
        cmin[y] := b[x,y];
  for y := 1 to n do
    for x := 1 to n do
      b[x,y] := b[x,y] - cmin[y]
end; (* reduction *)

procedure allocate;
var
  d : r2;
  finished : boolean;
  matches,matchno,oldalloc,x,y,temp : integer;

procedure change(row,col:integer);
var
  x : integer;
begin (* procedure change *)
  for x := 1 to n do
    begin
      d[x,col] := maxint;
      d[row,x] := maxint
      end
end; (* procedure change *)

begin (* procedure allocate *)
  for x := 1 to n do
for y := 1 to n do
begin
    d[x,y] := b[x,y];
    c[x,y] := 0
end;
matchno := 1;
finished := false;
oldalloc := 0;
alloc := 0;
while not finished do
begin
    for x := 1 to n do
begin
        matches := 0;
        for y := 1 to n do
            if d[x,y]=0
                then
                    begin
                        matches := matches + 1;
                        temp := y
                    end;
        if matches=matchno
            then
                begin
                    alloc := alloc + 1;
                    c[x,temp] := 1;
                    change(x,temp)
                end;
        if alloc=n
            then
                finished := true
    end;
if not finished
    then
for y := 1 to n do
begin
    matches := 0;
    for x := 1 to n do
        if d[x,y]=0
            then
                begin
                    matches := matches + 1;
                    temp := x
                end;
    if matches=matchno
        then
            begin
                alloc := alloc + 1;
                c[temp,y] := 1;
change(temp,y)
end;
if alloc=n
then
  finished := true
end;
if (oldalloc=alloc) and not finished
then
  begin
    matchno := matchno + 1;
    if matchno>(n-alloc)
      then
        finished := true
      end;
    oldalloc := alloc
  end
end; (* procedure allocate *)

procedure label1;
var
  rlabel,clabel : rl;
  rcheck : rb;
  checked : boolean;
  x,y,to_check : integer;

procedure check(row:integer);
var
  x,y : integer;
  found : boolean;

procedure breakthru(col:integer);
begin (* procedure breakthru *)
  while not checked do
    begin
      if rlabel[clabel[col]]=n+1
        then
          begin
            checked := true;
            alloc := alloc + 1;
            if alloc=n
              then
                done := true
            end
          else
            begin
              c[clabel[col],rlabel[clabel[col]]] := 0;
              col := rlabel[clabel[col]]
            end
      end
end

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
end
end; (* procedure breakthru *)

begin (* procedure check *)
x := 1;
while (x <= n) and not checked do
begin
if b[row,x]=0
then
if clabel[x]=0
then
begin
clabel[x] := row;
if not calloc[x]
then
breakthru(x)
else
begin
y := 1;
found := false;
while not found do
begin
if c[y,x]=1
then
begin
to_check := to_check + 1;
rlabelf[y] := x;
found := true
end;
y := y + 1
end
end
end
x := x + 1
end;
rcheck[row] := true;
to_check := to_check - 1
end; (* procedure check *)

procedure reduce;
var
x,y,min : integer;
begin (* procedure reduce *)
min := maxint;
for x := 1 to n do
for y := 1 to n do
if (rlabelf[x]<>0) and (clabel[y]=0)
then
if b[x,y]<min
then
    min := b[x,y];
for x := 1 to n do
for y := 1 to n do
begin
    if (rlabel[x]<0) and (clabel[y]=0) then
        b[x,y] := b[x,y] - min;
    if (rlabel[x]=0) and (clabel[y]<0) then
        b[x,y] := b[x,y] + min
end;

begin (* procedure reduce *)

to_check := 0;
checked := false;
for x := 1 to n do
begin
    rlabel[x] := 0;
    clabel[x] := 0;
    rcheck[x] := false;
    ralloc[x] := false;
    calloc[x] := false;
end;

for x := 1 to n do
for y := 1 to n do
if c[x,y]=1 then
begin
    ralloc[x] := true;
    calloc[y] := true
end;

x := 1;
while (x<=n) and not checked do
begin
    if not ralloc[x] then
begin
        to_check := to_check + 1;
        rlabel[x] := n + 1;
        check(x)
        end;
        x := x + 1
    end;
while (to_check>0) and not checked do
begin
    x := 1;
    while (x<=n) and not checked do

begin
  if (rlabel[x]<>0) and not rcheck[x]
  then
    check(x);
    x := x + 1
  end
end;
if not checked
then
  reduce
end; (* procedure label1 *)

procedure print;
var
  total,x,y : integer;
begin (* procedure print *)
  total := 0;
  for x := 1 to n do (* begin *)
    for y := 1 to n do (* begin *
      write((n + 1 - a[x,y]) * c[x,y]:3); *)
      total := total + (n + 1 - a[x,y]) * c[x,y];
    (* end; for y := 1 to n do *)
    write(n + 1 - a[x,y]:3);
    writeln
  end;
  writeln(*)
  write(f/'The seed is ',seed:3);
  gtot := gtot + total;
  writeln(f/' The total is ',total:5)
end; (* procedure print *)

begin (* main *)
assign(f/'hungtest');
rewrite(f);
writeln(f/'Hungarian method');
writeln(f/'size is ',n:3,' range is ',n:3);
writeln('enter starting seed');
readln(start);
gtot := 0;
for seed := start to (trials + start - 1) do
begin
  initialize;
  reduction;
  allocate;
  done := (alloc=n);
  while not done do
program main(input,output);

const
size = 13;
trials = 50;
type
ray = array[1..size,1..size] of integer;
ray1 = array[1..size] of integer;
var
start,rowi,coli,row,col,x,y,z,seed,maxrow,maxcol,total,gtotal : integer;
index,ciindex,colsec,riindex,rirowsec,ricoldiff,ricoldiff,rowmax,colmox : ray1;
a,b,c : ray;
rowalloc,colalloc : array[1..size] of boolean;
f : text;

procedure initialize;
var
x,y : integer;
begin (* procedure initialize *)
randseed := seed;
for x := 1 to size do
for y := 1 to size do
begin
  a[x,y] := random(size) + 1;
b[x,y] := a[x,y];
c[x,y] := 0;
rowalloc[x] := false;
colalloc[x] := false
end
end; (* procedure initialize *)

procedure print;
var
x,y : integer;
begin (* procedure print *)
total := 0;
for x := 1 to size do
for y := 1 to size do
  total := total + c[x,y] * b[x,y];
write(f,'The seed number is ',seed:4);
writeln(f', The total is ',total:5);
gtotal := gtotal + total
end; (* procedure print *)

procedure allocate(row,col:integer);
var
  x : integer;
begin (* procedure allocate *)
c[row,col] := 1;
for x := 1 to size do
begin
  a[row,x] := 0;
a[x,col] := 0
end;
rowalloc[row] := true;
colalloc[col] := true
end; (* procedure allocate *)

procedure solution;
var
  x,y,z : integer;
begin (* procedure solution *)
for z := 1 to size do
begin
  for x := 1 to size do
  begin
    rindex[x] := 0;
cindex[x] := 0;
rowmax[x] := 0;
rowsec[x] := 0;
colmax[x] := 0;
colsec[x] := 0;
rowdiff[x] := 0;
coldiff[x] := 0;
maxrow := -1;
maxcol := -1;
rowi := 0;
coli := 0
end;
for x := 1 to size do
for y := 1 to size do
begin
  if (a[x,y] > rowmax[x]) and (a[x,y] <> 0)
  then
    begin
      rowmax[x] := a[x,y];
cindex[x] := y
    end;
if (a[x,y] > colmax[y]) and (a[x,y] <> 0)
then
begin
    colmax[y] := a[x,y];
    rindex[y] := x
end
end;
for x := 1 to size do
for y := 1 to size do
begin
    if (a[x,y] > rowsec[x]) and (cindex[x] <> y)
    then
        rowsec[x] := a[x,y];
    if (a[x,y] > colsec[y]) and (rindex[y] <> x)
    then
        colsec[y] := a[x,y]
end;
for x := 1 to size do
begin
    rowdiff[x] := rowmax[x] - rowsec[x];
    coldiff[x] := colmax[x] - colsec[x]
end;
for x := 1 to size do
begin
    if (rowdiff[x] > maxrow) and (not rowalloc[x])
    then
        begin
            maxrow := rowdiff[x];
            rowi := x
        end;
    if (coldiff[x] > maxcol) and (not colalloc[x])
    then
        begin
            maxcol := coldiff[x];
            coli := x
        end
end;
if rowdiff[rowi] > coldiff[coli]
then
    allocate(rowi,cindex[rowi])
else
    allocate(rindex[coli],coli)
end;
(* procedure solution *)
begin (* main *)
assign(f,'vamtest');
rewrite(f);
writeln('enter starting seed');
writeln(f,'VAM method');
writeLn(f,'size:3', size:3); readLn(start);
gtotal := 0;
for seed := start to (trials + start - 1) do
begin
  initialize;
  solution;
  print
end;
writeLn(f,'The grand total for all trials is ',gtotal:6);
close(f)
end. (* main *)

Source Code for Greedy Method #1

program main(input,output);

const
  size = 10;
  trials = 50;
  start = 1;
type
  ray = array[1..size,1..size] of integer;
var
  done,foundone : boolean;
  match,row,col,cindex,rindex,x,y,ailoc : integer;
  seed,count,rowmax,colmmax,total,gtotal : integer;
  a,b,c : ray;
  f : text;

procedure initialize;
var
  x,y : integer;
begin (* procedure initialize *)
  randseed := seed;
  for x := 1 to size do
  for y := 1 to size do
    begin
      a[x,y] := random(size) + 1;
      b[x,y] := a[x,y];
      c[x,y] := 0
    end
end; (* procedure initialize *)

procedure print;
var
  x,y : integer;
begin (* procedure print *)
  total := 0;
"
for x := 1 to size do
for y := 1 to size do
  total := total + c[x,y] * b[x,y];
write(f,'The seed number is ',seed:4);
writeln(f,' The total is ',total:5);
gtotal := gtotal + total;
for x := 1 to size do
  begin
    for y := 1 to size do
      write(c[x,y]*b[x,y]:3,'');
      writeln
  end
end; (* procedure print *)

procedure allocate(row,col:integer);
var
  x : integer;
begin (* procedure allocate *)
  c[row,col] := 1;
  for x := 1 to size do
    begin
      a[row,x] := 0;
      a[x,col] := 0;
    end;
  foundone := true;
  alloc := alloc + 1;
  if alloc = size - 1
  then
    begin
      for x := 1 to size do
        for y := 1 to size do
          if a[x,y] > 0
          then
            c[x,y] := 1;
          done := true
    end
  end; (* procedure allocate *)

procedure solution;
var
  x,y,t : integer;
begin (* procedure solution *)
  done := false;
  t := 1;
  alloc := 0;
  match := size;
  while not done do
    begin
      foundone := false;
      for x := 1 to size do
...
for x := 1 to size do
begin
  rowmax := 0;
  for y := 1 to size do
    if a[x,y] > rowmax
    then
      begin
        rowmax := a[x,y];
        cindex := y
      end;
  count := 0;
  for y := 1 to size do
    if a[x,y] = rowmax
    then
      count := count + 1;
  if (count = t) and (rowmax = match)
  then
    begin
      writeln('call allocate with x = ',x:2/ and y = ',cindex:2);
      allocate(x,cindex)
    end
  end;
for y := 1 to size do
begin
  colmax := 0;
  for x := 1 to size do
    if a[x,y] > colmax
    then
      begin
        colmax := a[x,y];
        rindex := x
      end;
  count := 0;
  for x := 1 to size do
    if a[x,y] = colmax
    then
      count := count + 1;
  if (count = t) and (colmax = match)
  then
    begin
      writeln('call allocate with x = ',rindex:2/ and y = ',y:2);
      allocate(rindex,y)
    end
  end;
if not foundone
then
begin
  t := t + 1;
  if (t > size-alloc)
then
begin
match := match - 1;
t := 1
end
end
end; (* solution *)

begin (* main *)
assign(f,'greedy.tsO');
rewrite(f);
writeln(f,'greedy method #1');
writeln(f,'size is ',size:3,' range is ',size:3);
gtotal := 0;
for seed := start to (trials + start - 1) do
begin
initialize;
solution;
print
end;
writeln(f,'The grand total for all trials is ',gtotal:6);
close(f)
end. (* main *)

Source Code for Greedy Method #2

program main(input,output);

const
size = 10;
trials = 50;
start = 1;
type
ray = array[1..size,1..size] of integer;
var
done,foundone : boolean;
match,row,col,cindex,rindex,x,y,alloc : integer;
seed, count, rowmax, colmax, total, gtotal : integer;
a,b,c : ray;
f : text;

procedure initialize;
var
x,y : integer;
begin (* procedure initialize *)
randseed := seed;
for x := 1 to size do
for y := 1 to size do
begin
\(a[x,y] := \text{random} (\text{size}) + 1;\)
\(b[x,y] := a[x,y];\)
\(c[x,y] := 0\)
end
end; (* procedure initialize *)

procedure print;
var
\(x,y : \text{integer};\)
begin (* procedure print *)
\(\text{total} := 0;\)
for \(x := 1\) to \(\text{size}\) do
for \(y := 1\) to \(\text{size}\) do
\(\text{total} := \text{total} + c[x,y] * b[x,y];\)
write(f/'The seed number is ',seed:4);
writeln(f/'The total is ',total:5);
\(\text{gtotal} := \text{gtotal} + \text{total};\)
for \(x := 1\) to \(\text{size}\) do
begin
for \(y := 1\) to \(\text{size}\) do
write(c[x,y]*b[x,y]:3/');
writeln
end
end; (* procedure print *)

procedure allocate(row,col:integer);
var
\(x : \text{integer};\)
begin (* procedure allocate *)
\(c[\text{row},\text{col}] := 1;\)
for \(x := 1\) to \(\text{size}\) do
begin
\(a[\text{row},x] := 0;\)
\(a[x,\text{col}] := 0\)
end;
\(\text{foundone} := \text{true};\)
\(\text{alloc} := \text{alloc} + 1;\)
if \(\text{alloc} = \text{size} - 1\)
then
begin
for \(x := 1\) to \(\text{size}\) do
for \(y := 1\) to \(\text{size}\) do
if \(a[x,y] > 0\)
then
\(c[x,y] := 1;\)
\(\text{done} := \text{true}\)
end
end; (* procedure allocate *)
procedure solution;
    var
        x, y, t: integer;
    begin (* procedure solution *)
        done := false;
        t := 1;
        alloc := 0;
        while not done do
            begin
                foundone := false;
                for x := 1 to size do
                    begin
                        rowmax := 0;
                        for y := 1 to size do
                            if a[x, y] > rowmax
                                then
                                    begin
                                        rowmax := a[x, y];
                                        cindex := y
                                    end;
                            count := 0;
                            for y := 1 to size do
                                if a[x, y] = rowmax
                                    then
                                        count := count + 1;
                            if (count = t)
                                then
                                    writeln('call allocate with x = ',x:2,' and y = ',cindex:2);
                                    allocate(x,cindex)
                        end;
                    for y := 1 to size do
                        begin
                            colmax := 0;
                            for x := 1 to size do
                                if a[x, y] > colmax
                                    then
                                        begin
                                            colmax := a[x, y];
                                            rindex := x
                                        end;
                            count := 0;
                            for x := 1 to size do
                                if a[x, y] = colmax
                                    then
                                        count := count + 1;
                            if (count = t)
                                then
                                    begin
                                        end;
                                    end;
                            for y := 1 to size do
                                begin
                                    colmax := 0;
                                    for x := 1 to size do
                                        if a[x, y] > colmax
                                            then
                                                begin
                                                    colmax := a[x, y];
                                                    rindex := x
                                                end;
                                    count := 0;
                                    for x := 1 to size do
                                        if a[x, y] = colmax
                                            then
                                                count := count + 1;
                                    if (count = t)
                                        then
                                            begin
                                                end;
                                            end;
                                        end;
                                    end;
                                for y := 1 to size do
                                    begin
                                        colmax := 0;
                                        for x := 1 to size do
                                            if a[x, y] > colmax
                                                then
                                                    begin
                                                        colmax := a[x, y];
                                                        rindex := x
                                                    end;
                                        count := 0;
                                        for x := 1 to size do
                                            if a[x, y] = colmax
                                                then
                                                    count := count + 1;
                                        if (count = t)
                                            then
                                                begin
                                                    end;
                                                end;
                                            end;
                                        end;
                                    end;
                                for y := 1 to size do
                                    begin
                                        colmax := 0;
                                        for x := 1 to size do
                                            if a[x, y] > colmax
                                                then
                                                    begin
                                                        colmax := a[x, y];
                                                        rindex := x
                                                    end;
                                        count := 0;
                                        for x := 1 to size do
                                            if a[x, y] = colmax
                                                then
                                                    count := count + 1;
                                        if (count = t)
                                            then
                                                begin
                                                    end;
                                                end;
                                            end;
                                        end;
                                    end;
                                for y := 1 to size do
                                    begin
                                        colmax := 0;
                                        for x := 1 to size do
                                            if a[x, y] > colmax
                                                then
                                                    begin
                                                        colmax := a[x, y];
                                                        rindex := x
                                                    end;
                                        count := 0;
                                        for x := 1 to size do
                                            if a[x, y] = colmax
                                                then
                                                    count := count + 1;
                                        if (count = t)
                                            then
                                                begin
                                                    end;
                                                end;
                                            end;
                                        end;
                                    end;
                                for y := 1 to size do
                                    begin
                                        colmax := 0;
                                        for x := 1 to size do
                                            if a[x, y] > colmax
                                                then
                                                    begin
                                                        colmax := a[x, y];
                                                        rindex := x
                                                    end;
                                        count := 0;
                                        for x := 1 to size do
                                            if a[x, y] = colmax
                                                then
                                                    count := count + 1;
                                        if (count = t)
                                            then
                                                begin
                                                    end;
                                                end;
                                            end;
                                        end;
                                    end;
                                for y := 1 to size do
                                    begin
                                        colmax := 0;
                                        for x := 1 to size do
                                            if a[x, y] > colmax
                                                then
                                                    begin
                                                        colmax := a[x, y];
                                                        rindex := x
                                                    end;
                                        count := 0;
                                        for x := 1 to size do
                                            if a[x, y] = colmax
                                                then
                                                    count := count + 1;
                                        if (count = t)
                                            then
                                                begin
                                                    end;
                                                end;
                                            end;
                                        end;
                                    end;
                                for y := 1 to size do
                                    begin
                                        colmax := 0;
                                        for x := 1 to size do
                                            if a[x, y] > colmax
                                                then
                                                    begin
                                                        colmax := a[x, y];
                                                        rindex := x
                                                    end;
                                        count := 0;
                                        for x := 1 to size do
                                            if a[x, y] = colmax
                                                then
                                                    count := count + 1;
                                        if (count = t)
                                            then
                                                begin
                                                    end;
                                                end;
                                            end;
                                        end;
                                    end;
                                for y := 1 to size do
                                    begin
                                        colmax := 0;
                                        for x := 1 to size do
                                            if a[x, y] > colmax
                                                then
                                                    begin
                                                        colmax := a[x, y];
                                                        rindex := x
                                                    end;
                                        count := 0;
                                        for x := 1 to size do
                                            if a[x, y] = colmax
                                                then
                                                    count := count + 1;
                                        if (count = t)
                                            then
                                                begin
                                                    end;
                                                end;
                                            end;
                                        end;
                                    end;
                                for y := 1 to size do
                                    begin
                                        colmax := 0;
                                        for x := 1 to size do
                                            if a[x, y] > colmax
                                                then
                                                    begin
                                                        colmax := a[x, y];
                                                        rindex := x
                                                    end;
                                        count := 0;
                                        for x := 1 to size do
                                            if a[x, y] = colmax
                                                then
                                                    count := count + 1;
                                        if (count = t)
begin
  writeln('call allocate with x = ',rindex:2,' and y = ',y:2);
  allocate(rindex,y)
end end;
if not foundone then
t := t + 1
end end; (* solution *)

begin (* main *)
assign(f,'greedy1.tst');
rewrite(f);
writeln(f,'greedy method #2');
writeln(f,'size is ',size:3,' range is ',size:3);
gtotal := 0;
for seed := start to (trials + start - 1) do
begin
  initialize;
  solution;
  print
end;
writeln(f,'The grand total for all trials is ',gtotal:6);
close(f)
end. (* main *)
<table>
<thead>
<tr>
<th>Seed Number</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>83</td>
</tr>
<tr>
<td>2</td>
<td>86</td>
</tr>
<tr>
<td>3</td>
<td>88</td>
</tr>
<tr>
<td>4</td>
<td>83</td>
</tr>
<tr>
<td>5</td>
<td>88</td>
</tr>
<tr>
<td>6</td>
<td>90</td>
</tr>
<tr>
<td>7</td>
<td>93</td>
</tr>
<tr>
<td>8</td>
<td>76</td>
</tr>
<tr>
<td>9</td>
<td>89</td>
</tr>
<tr>
<td>10</td>
<td>88</td>
</tr>
<tr>
<td>11</td>
<td>82</td>
</tr>
<tr>
<td>12</td>
<td>87</td>
</tr>
<tr>
<td>13</td>
<td>89</td>
</tr>
<tr>
<td>14</td>
<td>89</td>
</tr>
<tr>
<td>15</td>
<td>91</td>
</tr>
<tr>
<td>16</td>
<td>89</td>
</tr>
<tr>
<td>17</td>
<td>86</td>
</tr>
<tr>
<td>18</td>
<td>91</td>
</tr>
<tr>
<td>19</td>
<td>96</td>
</tr>
<tr>
<td>20</td>
<td>94</td>
</tr>
<tr>
<td>21</td>
<td>87</td>
</tr>
<tr>
<td>22</td>
<td>89</td>
</tr>
<tr>
<td>23</td>
<td>80</td>
</tr>
<tr>
<td>24</td>
<td>81</td>
</tr>
<tr>
<td>25</td>
<td>77</td>
</tr>
<tr>
<td>26</td>
<td>83</td>
</tr>
<tr>
<td>27</td>
<td>84</td>
</tr>
<tr>
<td>28</td>
<td>84</td>
</tr>
<tr>
<td>29</td>
<td>79</td>
</tr>
<tr>
<td>30</td>
<td>86</td>
</tr>
<tr>
<td>31</td>
<td>83</td>
</tr>
<tr>
<td>32</td>
<td>87</td>
</tr>
<tr>
<td>33</td>
<td>88</td>
</tr>
<tr>
<td>34</td>
<td>88</td>
</tr>
<tr>
<td>35</td>
<td>88</td>
</tr>
<tr>
<td>36</td>
<td>85</td>
</tr>
<tr>
<td>37</td>
<td>85</td>
</tr>
<tr>
<td>38</td>
<td>93</td>
</tr>
<tr>
<td>39</td>
<td>89</td>
</tr>
<tr>
<td>40</td>
<td>87</td>
</tr>
<tr>
<td>41</td>
<td>91</td>
</tr>
<tr>
<td>42</td>
<td>86</td>
</tr>
<tr>
<td>43</td>
<td>81</td>
</tr>
<tr>
<td>44</td>
<td>91</td>
</tr>
<tr>
<td>45</td>
<td>86</td>
</tr>
<tr>
<td>46</td>
<td>89</td>
</tr>
<tr>
<td>47</td>
<td>83</td>
</tr>
<tr>
<td>48</td>
<td>93</td>
</tr>
<tr>
<td>49</td>
<td>84</td>
</tr>
<tr>
<td>50</td>
<td>85</td>
</tr>
<tr>
<td>51</td>
<td>84</td>
</tr>
<tr>
<td>52</td>
<td>86</td>
</tr>
<tr>
<td>53</td>
<td>86</td>
</tr>
<tr>
<td>54</td>
<td>86</td>
</tr>
<tr>
<td>55</td>
<td>82</td>
</tr>
<tr>
<td>56</td>
<td>90</td>
</tr>
<tr>
<td>57</td>
<td>78</td>
</tr>
<tr>
<td>58</td>
<td>89</td>
</tr>
<tr>
<td>59</td>
<td>85</td>
</tr>
<tr>
<td>60</td>
<td>84</td>
</tr>
<tr>
<td>61</td>
<td>76</td>
</tr>
<tr>
<td>62</td>
<td>88</td>
</tr>
<tr>
<td>63</td>
<td>84</td>
</tr>
<tr>
<td>64</td>
<td>87</td>
</tr>
<tr>
<td>65</td>
<td>82</td>
</tr>
<tr>
<td>66</td>
<td>80</td>
</tr>
<tr>
<td>67</td>
<td>80</td>
</tr>
<tr>
<td>68</td>
<td>85</td>
</tr>
<tr>
<td>69</td>
<td>85</td>
</tr>
<tr>
<td>70</td>
<td>89</td>
</tr>
<tr>
<td>71</td>
<td>90</td>
</tr>
<tr>
<td>72</td>
<td>88</td>
</tr>
<tr>
<td>73</td>
<td>80</td>
</tr>
<tr>
<td>74</td>
<td>90</td>
</tr>
<tr>
<td>75</td>
<td>91</td>
</tr>
<tr>
<td>76</td>
<td>85</td>
</tr>
<tr>
<td>77</td>
<td>94</td>
</tr>
<tr>
<td>78</td>
<td>83</td>
</tr>
<tr>
<td>79</td>
<td>93</td>
</tr>
<tr>
<td>80</td>
<td>86</td>
</tr>
<tr>
<td>81</td>
<td>83</td>
</tr>
<tr>
<td>82</td>
<td>87</td>
</tr>
<tr>
<td>83</td>
<td>81</td>
</tr>
<tr>
<td>84</td>
<td>87</td>
</tr>
<tr>
<td>85</td>
<td>97</td>
</tr>
<tr>
<td>86</td>
<td>90</td>
</tr>
<tr>
<td>87</td>
<td>92</td>
</tr>
<tr>
<td>88</td>
<td>93</td>
</tr>
<tr>
<td>89</td>
<td>94</td>
</tr>
<tr>
<td>90</td>
<td>86</td>
</tr>
<tr>
<td>91</td>
<td>80</td>
</tr>
<tr>
<td>92</td>
<td>80</td>
</tr>
<tr>
<td>93</td>
<td>91</td>
</tr>
<tr>
<td>94</td>
<td>82</td>
</tr>
<tr>
<td>95</td>
<td>83</td>
</tr>
<tr>
<td>96</td>
<td>86</td>
</tr>
<tr>
<td>97</td>
<td>86</td>
</tr>
<tr>
<td>98</td>
<td>87</td>
</tr>
<tr>
<td>99</td>
<td>85</td>
</tr>
<tr>
<td>100</td>
<td>85</td>
</tr>
<tr>
<td>Grand Total</td>
<td>4330</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Seed Number</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>51</td>
<td>84</td>
</tr>
<tr>
<td>52</td>
<td>86</td>
</tr>
<tr>
<td>53</td>
<td>86</td>
</tr>
<tr>
<td>54</td>
<td>86</td>
</tr>
<tr>
<td>55</td>
<td>82</td>
</tr>
<tr>
<td>56</td>
<td>90</td>
</tr>
<tr>
<td>57</td>
<td>78</td>
</tr>
<tr>
<td>58</td>
<td>89</td>
</tr>
<tr>
<td>59</td>
<td>85</td>
</tr>
<tr>
<td>60</td>
<td>84</td>
</tr>
<tr>
<td>61</td>
<td>76</td>
</tr>
<tr>
<td>62</td>
<td>88</td>
</tr>
<tr>
<td>63</td>
<td>84</td>
</tr>
<tr>
<td>64</td>
<td>87</td>
</tr>
<tr>
<td>65</td>
<td>82</td>
</tr>
<tr>
<td>66</td>
<td>80</td>
</tr>
<tr>
<td>67</td>
<td>86</td>
</tr>
<tr>
<td>68</td>
<td>85</td>
</tr>
<tr>
<td>69</td>
<td>85</td>
</tr>
<tr>
<td>70</td>
<td>89</td>
</tr>
<tr>
<td>71</td>
<td>90</td>
</tr>
<tr>
<td>72</td>
<td>88</td>
</tr>
<tr>
<td>73</td>
<td>80</td>
</tr>
<tr>
<td>74</td>
<td>90</td>
</tr>
<tr>
<td>75</td>
<td>91</td>
</tr>
<tr>
<td>76</td>
<td>85</td>
</tr>
<tr>
<td>77</td>
<td>94</td>
</tr>
<tr>
<td>78</td>
<td>83</td>
</tr>
<tr>
<td>79</td>
<td>93</td>
</tr>
<tr>
<td>80</td>
<td>86</td>
</tr>
<tr>
<td>81</td>
<td>83</td>
</tr>
<tr>
<td>82</td>
<td>87</td>
</tr>
<tr>
<td>83</td>
<td>81</td>
</tr>
<tr>
<td>84</td>
<td>87</td>
</tr>
<tr>
<td>85</td>
<td>97</td>
</tr>
<tr>
<td>86</td>
<td>90</td>
</tr>
<tr>
<td>87</td>
<td>92</td>
</tr>
<tr>
<td>88</td>
<td>93</td>
</tr>
<tr>
<td>89</td>
<td>94</td>
</tr>
<tr>
<td>90</td>
<td>86</td>
</tr>
<tr>
<td>91</td>
<td>80</td>
</tr>
<tr>
<td>92</td>
<td>80</td>
</tr>
<tr>
<td>93</td>
<td>91</td>
</tr>
<tr>
<td>94</td>
<td>82</td>
</tr>
<tr>
<td>95</td>
<td>83</td>
</tr>
<tr>
<td>96</td>
<td>86</td>
</tr>
<tr>
<td>97</td>
<td>86</td>
</tr>
<tr>
<td>98</td>
<td>87</td>
</tr>
<tr>
<td>99</td>
<td>85</td>
</tr>
<tr>
<td>100</td>
<td>85</td>
</tr>
<tr>
<td>Grand Total</td>
<td>4292</td>
</tr>
</tbody>
</table>
# Results with Greedy Method #2

**greedy method #2**  
size is 10 range is 10

<table>
<thead>
<tr>
<th>The seed number is</th>
<th>The total is</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>82</td>
</tr>
<tr>
<td>2</td>
<td>92</td>
</tr>
<tr>
<td>3</td>
<td>93</td>
</tr>
<tr>
<td>4</td>
<td>86</td>
</tr>
<tr>
<td>5</td>
<td>92</td>
</tr>
<tr>
<td>6</td>
<td>94</td>
</tr>
<tr>
<td>7</td>
<td>90</td>
</tr>
<tr>
<td>8</td>
<td>78</td>
</tr>
<tr>
<td>9</td>
<td>89</td>
</tr>
<tr>
<td>10</td>
<td>82</td>
</tr>
<tr>
<td>11</td>
<td>89</td>
</tr>
<tr>
<td>12</td>
<td>90</td>
</tr>
<tr>
<td>13</td>
<td>86</td>
</tr>
<tr>
<td>14</td>
<td>89</td>
</tr>
<tr>
<td>15</td>
<td>91</td>
</tr>
<tr>
<td>16</td>
<td>89</td>
</tr>
<tr>
<td>17</td>
<td>87</td>
</tr>
<tr>
<td>18</td>
<td>90</td>
</tr>
<tr>
<td>19</td>
<td>90</td>
</tr>
<tr>
<td>20</td>
<td>92</td>
</tr>
<tr>
<td>21</td>
<td>86</td>
</tr>
<tr>
<td>22</td>
<td>89</td>
</tr>
<tr>
<td>23</td>
<td>85</td>
</tr>
<tr>
<td>24</td>
<td>85</td>
</tr>
<tr>
<td>25</td>
<td>77</td>
</tr>
<tr>
<td>26</td>
<td>81</td>
</tr>
<tr>
<td>27</td>
<td>94</td>
</tr>
<tr>
<td>28</td>
<td>88</td>
</tr>
<tr>
<td>29</td>
<td>78</td>
</tr>
<tr>
<td>30</td>
<td>86</td>
</tr>
<tr>
<td>31</td>
<td>93</td>
</tr>
<tr>
<td>32</td>
<td>86</td>
</tr>
<tr>
<td>33</td>
<td>86</td>
</tr>
<tr>
<td>34</td>
<td>85</td>
</tr>
<tr>
<td>35</td>
<td>85</td>
</tr>
<tr>
<td>36</td>
<td>81</td>
</tr>
<tr>
<td>37</td>
<td>83</td>
</tr>
<tr>
<td>38</td>
<td>93</td>
</tr>
<tr>
<td>39</td>
<td>90</td>
</tr>
<tr>
<td>40</td>
<td>83</td>
</tr>
<tr>
<td>41</td>
<td>90</td>
</tr>
<tr>
<td>42</td>
<td>80</td>
</tr>
<tr>
<td>43</td>
<td>84</td>
</tr>
<tr>
<td>44</td>
<td>89</td>
</tr>
<tr>
<td>45</td>
<td>85</td>
</tr>
<tr>
<td>46</td>
<td>93</td>
</tr>
<tr>
<td>47</td>
<td>87</td>
</tr>
<tr>
<td>48</td>
<td>89</td>
</tr>
<tr>
<td>49</td>
<td>81</td>
</tr>
<tr>
<td>50</td>
<td>84</td>
</tr>
</tbody>
</table>

The grand total for all trials is 4355

---

**greedy method #2**  
size is 10 range is 10

<table>
<thead>
<tr>
<th>The seed number is</th>
<th>The total is</th>
</tr>
</thead>
<tbody>
<tr>
<td>51</td>
<td>82</td>
</tr>
<tr>
<td>52</td>
<td>89</td>
</tr>
<tr>
<td>53</td>
<td>84</td>
</tr>
<tr>
<td>54</td>
<td>84</td>
</tr>
<tr>
<td>55</td>
<td>80</td>
</tr>
<tr>
<td>56</td>
<td>89</td>
</tr>
<tr>
<td>57</td>
<td>79</td>
</tr>
<tr>
<td>58</td>
<td>92</td>
</tr>
<tr>
<td>59</td>
<td>81</td>
</tr>
<tr>
<td>60</td>
<td>84</td>
</tr>
<tr>
<td>61</td>
<td>79</td>
</tr>
<tr>
<td>62</td>
<td>88</td>
</tr>
<tr>
<td>63</td>
<td>84</td>
</tr>
<tr>
<td>64</td>
<td>84</td>
</tr>
<tr>
<td>65</td>
<td>84</td>
</tr>
<tr>
<td>66</td>
<td>87</td>
</tr>
<tr>
<td>67</td>
<td>84</td>
</tr>
<tr>
<td>68</td>
<td>84</td>
</tr>
<tr>
<td>69</td>
<td>90</td>
</tr>
<tr>
<td>70</td>
<td>86</td>
</tr>
<tr>
<td>71</td>
<td>84</td>
</tr>
<tr>
<td>72</td>
<td>82</td>
</tr>
<tr>
<td>73</td>
<td>74</td>
</tr>
<tr>
<td>74</td>
<td>85</td>
</tr>
<tr>
<td>75</td>
<td>95</td>
</tr>
<tr>
<td>76</td>
<td>87</td>
</tr>
<tr>
<td>77</td>
<td>95</td>
</tr>
<tr>
<td>78</td>
<td>97</td>
</tr>
<tr>
<td>79</td>
<td>94</td>
</tr>
<tr>
<td>80</td>
<td>89</td>
</tr>
<tr>
<td>81</td>
<td>86</td>
</tr>
<tr>
<td>82</td>
<td>85</td>
</tr>
<tr>
<td>83</td>
<td>85</td>
</tr>
<tr>
<td>84</td>
<td>87</td>
</tr>
<tr>
<td>85</td>
<td>92</td>
</tr>
<tr>
<td>86</td>
<td>84</td>
</tr>
<tr>
<td>87</td>
<td>92</td>
</tr>
<tr>
<td>88</td>
<td>85</td>
</tr>
<tr>
<td>89</td>
<td>89</td>
</tr>
<tr>
<td>90</td>
<td>86</td>
</tr>
<tr>
<td>91</td>
<td>90</td>
</tr>
<tr>
<td>92</td>
<td>80</td>
</tr>
<tr>
<td>93</td>
<td>93</td>
</tr>
<tr>
<td>94</td>
<td>80</td>
</tr>
<tr>
<td>95</td>
<td>95</td>
</tr>
<tr>
<td>96</td>
<td>87</td>
</tr>
<tr>
<td>97</td>
<td>92</td>
</tr>
<tr>
<td>98</td>
<td>87</td>
</tr>
<tr>
<td>99</td>
<td>92</td>
</tr>
<tr>
<td>100</td>
<td>82</td>
</tr>
</tbody>
</table>

The grand total for all trials is 4286

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Results with VAM

<table>
<thead>
<tr>
<th>Seed Number</th>
<th>Total</th>
<th>Seed Number</th>
<th>Total</th>
<th>Seed Number</th>
<th>Total</th>
<th>Seed Number</th>
<th>Total</th>
<th>Seed Number</th>
<th>Total</th>
<th>Seed Number</th>
<th>Total</th>
<th>Seed Number</th>
<th>Total</th>
<th>Seed Number</th>
<th>Total</th>
<th>Seed Number</th>
<th>Total</th>
<th>Seed Number</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>150</td>
<td>2</td>
<td>155</td>
<td>3</td>
<td>148</td>
<td>4</td>
<td>150</td>
<td>5</td>
<td>162</td>
<td>6</td>
<td>159</td>
<td>7</td>
<td>156</td>
<td>8</td>
<td>153</td>
<td>9</td>
<td>156</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>156</td>
<td>11</td>
<td>156</td>
<td>12</td>
<td>150</td>
<td>13</td>
<td>150</td>
<td>14</td>
<td>156</td>
<td>15</td>
<td>157</td>
<td>16</td>
<td>154</td>
<td>17</td>
<td>158</td>
<td>18</td>
<td>154</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>149</td>
<td>20</td>
<td>161</td>
<td>21</td>
<td>149</td>
<td>22</td>
<td>154</td>
<td>23</td>
<td>154</td>
<td>24</td>
<td>154</td>
<td>25</td>
<td>157</td>
<td>26</td>
<td>149</td>
<td>27</td>
<td>149</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>157</td>
<td>29</td>
<td>155</td>
<td>30</td>
<td>160</td>
<td>31</td>
<td>156</td>
<td>32</td>
<td>157</td>
<td>33</td>
<td>141</td>
<td>34</td>
<td>152</td>
<td>35</td>
<td>156</td>
<td>36</td>
<td>158</td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>152</td>
<td>38</td>
<td>154</td>
<td>39</td>
<td>157</td>
<td>40</td>
<td>153</td>
<td>41</td>
<td>149</td>
<td>42</td>
<td>149</td>
<td>43</td>
<td>153</td>
<td>44</td>
<td>154</td>
<td>45</td>
<td>154</td>
<td></td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>154</td>
<td>47</td>
<td>160</td>
<td>48</td>
<td>155</td>
<td>49</td>
<td>151</td>
<td>50</td>
<td>152</td>
<td>51</td>
<td>153</td>
<td>52</td>
<td>156</td>
<td>53</td>
<td>144</td>
<td>54</td>
<td>154</td>
<td></td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>148</td>
<td>56</td>
<td>157</td>
<td>57</td>
<td>156</td>
<td>58</td>
<td>157</td>
<td>59</td>
<td>154</td>
<td>60</td>
<td>146</td>
<td>61</td>
<td>153</td>
<td>62</td>
<td>157</td>
<td>63</td>
<td>158</td>
<td></td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>157</td>
<td>65</td>
<td>151</td>
<td>66</td>
<td>161</td>
<td>67</td>
<td>150</td>
<td>68</td>
<td>156</td>
<td>69</td>
<td>154</td>
<td>70</td>
<td>151</td>
<td>71</td>
<td>158</td>
<td>72</td>
<td>151</td>
<td></td>
<td></td>
</tr>
<tr>
<td>73</td>
<td>151</td>
<td>74</td>
<td>160</td>
<td>75</td>
<td>155</td>
<td>76</td>
<td>158</td>
<td>77</td>
<td>163</td>
<td>78</td>
<td>150</td>
<td>79</td>
<td>151</td>
<td>80</td>
<td>163</td>
<td>81</td>
<td>157</td>
<td></td>
<td></td>
</tr>
<tr>
<td>82</td>
<td>156</td>
<td>83</td>
<td>148</td>
<td>84</td>
<td>151</td>
<td>85</td>
<td>156</td>
<td>86</td>
<td>154</td>
<td>87</td>
<td>155</td>
<td>88</td>
<td>156</td>
<td>89</td>
<td>154</td>
<td>90</td>
<td>152</td>
<td></td>
<td></td>
</tr>
<tr>
<td>91</td>
<td>151</td>
<td>92</td>
<td>157</td>
<td>93</td>
<td>160</td>
<td>94</td>
<td>161</td>
<td>95</td>
<td>154</td>
<td>96</td>
<td>151</td>
<td>97</td>
<td>152</td>
<td>98</td>
<td>157</td>
<td>99</td>
<td>153</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>151</td>
<td>101</td>
<td>152</td>
<td>102</td>
<td>7699</td>
<td>103</td>
<td></td>
<td>104</td>
<td></td>
<td>105</td>
<td></td>
<td>106</td>
<td></td>
<td>107</td>
<td></td>
<td>108</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The grand total for all trials is 7699

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
| Seed Number | Total 1  | Total 2  | Total 3  | Total 4  | Total 5  | Total 6  | Total 7  | Total 8  | Total 9  | Total 10 | Total 11 | Total 12 | Total 13 | Total 14 | Total 15 | Total 16 | Total 17 | Total 18 | Total 19 | Total 20 |
|-------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| 101         | 152     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 102         | 159     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 103         | 149     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 104         | 150     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 105         | 159     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 106         | 159     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 107         | 155     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 108         | 159     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 109         | 143     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 110         | 152     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 111         | 153     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 112         | 157     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 113         | 160     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 114         | 155     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 115         | 140     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 116         | 155     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 117         | 154     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 118         | 154     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 119         | 164     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 120         | 155     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 121         | 156     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 122         | 157     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 123         | 151     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 124         | 162     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 125         | 159     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 126         | 158     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 127         | 148     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 128         | 152     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 129         | 160     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 130         | 155     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 131         | 157     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 132         | 148     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 133         | 156     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 134         | 149     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 135         | 148     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 136         | 155     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 137         | 159     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 138         | 158     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 139         | 161     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 140         | 152     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 141         | 148     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 142         | 152     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 143         | 152     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 144         | 160     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 145         | 157     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 146         | 157     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 147         | 158     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 148         | 158     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 149         | 154     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| 150         | 152     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |

The grand total for all trials is 7733
<table>
<thead>
<tr>
<th>Seed Number</th>
<th>Total Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>201</td>
<td>309</td>
</tr>
<tr>
<td>202</td>
<td>308</td>
</tr>
<tr>
<td>203</td>
<td>297</td>
</tr>
<tr>
<td>204</td>
<td>278</td>
</tr>
<tr>
<td>205</td>
<td>313</td>
</tr>
<tr>
<td>206</td>
<td>310</td>
</tr>
<tr>
<td>207</td>
<td>289</td>
</tr>
<tr>
<td>208</td>
<td>306</td>
</tr>
<tr>
<td>209</td>
<td>315</td>
</tr>
<tr>
<td>210</td>
<td>298</td>
</tr>
<tr>
<td>211</td>
<td>312</td>
</tr>
<tr>
<td>212</td>
<td>311</td>
</tr>
<tr>
<td>213</td>
<td>299</td>
</tr>
<tr>
<td>214</td>
<td>303</td>
</tr>
<tr>
<td>215</td>
<td>309</td>
</tr>
<tr>
<td>216</td>
<td>304</td>
</tr>
<tr>
<td>217</td>
<td>306</td>
</tr>
<tr>
<td>218</td>
<td>305</td>
</tr>
<tr>
<td>219</td>
<td>312</td>
</tr>
<tr>
<td>220</td>
<td>312</td>
</tr>
<tr>
<td>221</td>
<td>309</td>
</tr>
<tr>
<td>222</td>
<td>316</td>
</tr>
<tr>
<td>223</td>
<td>301</td>
</tr>
<tr>
<td>224</td>
<td>309</td>
</tr>
<tr>
<td>225</td>
<td>302</td>
</tr>
<tr>
<td>226</td>
<td>305</td>
</tr>
<tr>
<td>227</td>
<td>303</td>
</tr>
<tr>
<td>228</td>
<td>325</td>
</tr>
<tr>
<td>229</td>
<td>309</td>
</tr>
<tr>
<td>230</td>
<td>310</td>
</tr>
<tr>
<td>231</td>
<td>287</td>
</tr>
<tr>
<td>232</td>
<td>315</td>
</tr>
<tr>
<td>233</td>
<td>279</td>
</tr>
<tr>
<td>234</td>
<td>305</td>
</tr>
<tr>
<td>235</td>
<td>283</td>
</tr>
<tr>
<td>236</td>
<td>305</td>
</tr>
<tr>
<td>237</td>
<td>305</td>
</tr>
<tr>
<td>238</td>
<td>314</td>
</tr>
<tr>
<td>239</td>
<td>297</td>
</tr>
<tr>
<td>240</td>
<td>314</td>
</tr>
<tr>
<td>241</td>
<td>296</td>
</tr>
<tr>
<td>242</td>
<td>317</td>
</tr>
<tr>
<td>243</td>
<td>305</td>
</tr>
<tr>
<td>244</td>
<td>296</td>
</tr>
<tr>
<td>245</td>
<td>283</td>
</tr>
<tr>
<td>246</td>
<td>304</td>
</tr>
<tr>
<td>247</td>
<td>311</td>
</tr>
<tr>
<td>248</td>
<td>298</td>
</tr>
<tr>
<td>249</td>
<td>301</td>
</tr>
<tr>
<td>250</td>
<td>307</td>
</tr>
</tbody>
</table>

Grand total for all trials is 15225
<table>
<thead>
<tr>
<th>Seed Number</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>401</td>
<td>378</td>
</tr>
<tr>
<td>402</td>
<td>382</td>
</tr>
<tr>
<td>403</td>
<td>370</td>
</tr>
<tr>
<td>404</td>
<td>367</td>
</tr>
<tr>
<td>405</td>
<td>378</td>
</tr>
<tr>
<td>406</td>
<td>380</td>
</tr>
<tr>
<td>407</td>
<td>373</td>
</tr>
<tr>
<td>408</td>
<td>359</td>
</tr>
<tr>
<td>409</td>
<td>370</td>
</tr>
<tr>
<td>410</td>
<td>369</td>
</tr>
<tr>
<td>411</td>
<td>371</td>
</tr>
<tr>
<td>412</td>
<td>392</td>
</tr>
<tr>
<td>413</td>
<td>361</td>
</tr>
<tr>
<td>414</td>
<td>374</td>
</tr>
<tr>
<td>415</td>
<td>370</td>
</tr>
<tr>
<td>416</td>
<td>357</td>
</tr>
<tr>
<td>417</td>
<td>379</td>
</tr>
<tr>
<td>418</td>
<td>377</td>
</tr>
<tr>
<td>419</td>
<td>384</td>
</tr>
<tr>
<td>420</td>
<td>374</td>
</tr>
<tr>
<td>421</td>
<td>379</td>
</tr>
<tr>
<td>422</td>
<td>378</td>
</tr>
<tr>
<td>423</td>
<td>381</td>
</tr>
<tr>
<td>424</td>
<td>379</td>
</tr>
<tr>
<td>425</td>
<td>375</td>
</tr>
<tr>
<td>426</td>
<td>385</td>
</tr>
<tr>
<td>427</td>
<td>371</td>
</tr>
<tr>
<td>428</td>
<td>379</td>
</tr>
<tr>
<td>429</td>
<td>380</td>
</tr>
<tr>
<td>430</td>
<td>384</td>
</tr>
<tr>
<td>431</td>
<td>364</td>
</tr>
<tr>
<td>432</td>
<td>377</td>
</tr>
<tr>
<td>433</td>
<td>373</td>
</tr>
<tr>
<td>434</td>
<td>377</td>
</tr>
<tr>
<td>435</td>
<td>375</td>
</tr>
<tr>
<td>436</td>
<td>388</td>
</tr>
<tr>
<td>437</td>
<td>376</td>
</tr>
<tr>
<td>438</td>
<td>376</td>
</tr>
<tr>
<td>439</td>
<td>381</td>
</tr>
<tr>
<td>440</td>
<td>371</td>
</tr>
<tr>
<td>441</td>
<td>387</td>
</tr>
<tr>
<td>442</td>
<td>376</td>
</tr>
<tr>
<td>443</td>
<td>379</td>
</tr>
<tr>
<td>444</td>
<td>384</td>
</tr>
<tr>
<td>445</td>
<td>380</td>
</tr>
<tr>
<td>446</td>
<td>380</td>
</tr>
<tr>
<td>447</td>
<td>368</td>
</tr>
<tr>
<td>448</td>
<td>378</td>
</tr>
<tr>
<td>449</td>
<td>371</td>
</tr>
<tr>
<td>450</td>
<td>380</td>
</tr>
<tr>
<td>451</td>
<td>381</td>
</tr>
<tr>
<td>452</td>
<td>377</td>
</tr>
<tr>
<td>453</td>
<td>381</td>
</tr>
<tr>
<td>454</td>
<td>381</td>
</tr>
<tr>
<td>455</td>
<td>378</td>
</tr>
<tr>
<td>456</td>
<td>379</td>
</tr>
<tr>
<td>457</td>
<td>382</td>
</tr>
<tr>
<td>458</td>
<td>373</td>
</tr>
<tr>
<td>459</td>
<td>374</td>
</tr>
<tr>
<td>460</td>
<td>366</td>
</tr>
<tr>
<td>461</td>
<td>388</td>
</tr>
<tr>
<td>462</td>
<td>379</td>
</tr>
<tr>
<td>463</td>
<td>378</td>
</tr>
<tr>
<td>464</td>
<td>370</td>
</tr>
<tr>
<td>465</td>
<td>374</td>
</tr>
<tr>
<td>466</td>
<td>366</td>
</tr>
<tr>
<td>467</td>
<td>373</td>
</tr>
<tr>
<td>468</td>
<td>381</td>
</tr>
<tr>
<td>469</td>
<td>379</td>
</tr>
<tr>
<td>470</td>
<td>379</td>
</tr>
<tr>
<td>471</td>
<td>375</td>
</tr>
<tr>
<td>472</td>
<td>370</td>
</tr>
<tr>
<td>473</td>
<td>380</td>
</tr>
<tr>
<td>474</td>
<td>359</td>
</tr>
<tr>
<td>475</td>
<td>372</td>
</tr>
<tr>
<td>476</td>
<td>370</td>
</tr>
<tr>
<td>477</td>
<td>368</td>
</tr>
<tr>
<td>478</td>
<td>385</td>
</tr>
<tr>
<td>479</td>
<td>370</td>
</tr>
<tr>
<td>480</td>
<td>382</td>
</tr>
<tr>
<td>481</td>
<td>378</td>
</tr>
<tr>
<td>482</td>
<td>388</td>
</tr>
<tr>
<td>483</td>
<td>380</td>
</tr>
<tr>
<td>484</td>
<td>384</td>
</tr>
<tr>
<td>485</td>
<td>384</td>
</tr>
<tr>
<td>486</td>
<td>384</td>
</tr>
<tr>
<td>487</td>
<td>376</td>
</tr>
<tr>
<td>488</td>
<td>383</td>
</tr>
<tr>
<td>489</td>
<td>368</td>
</tr>
<tr>
<td>490</td>
<td>375</td>
</tr>
<tr>
<td>491</td>
<td>381</td>
</tr>
<tr>
<td>492</td>
<td>373</td>
</tr>
<tr>
<td>493</td>
<td>377</td>
</tr>
<tr>
<td>494</td>
<td>379</td>
</tr>
<tr>
<td>495</td>
<td>369</td>
</tr>
<tr>
<td>496</td>
<td>333</td>
</tr>
<tr>
<td>497</td>
<td>386</td>
</tr>
<tr>
<td>498</td>
<td>374</td>
</tr>
<tr>
<td>499</td>
<td>385</td>
</tr>
<tr>
<td>500</td>
<td>374</td>
</tr>
<tr>
<td>Grand Total</td>
<td>18817</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Seed Number</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>451</td>
<td>381</td>
</tr>
<tr>
<td>452</td>
<td>377</td>
</tr>
<tr>
<td>453</td>
<td>381</td>
</tr>
<tr>
<td>454</td>
<td>381</td>
</tr>
<tr>
<td>455</td>
<td>378</td>
</tr>
<tr>
<td>456</td>
<td>379</td>
</tr>
<tr>
<td>457</td>
<td>382</td>
</tr>
<tr>
<td>458</td>
<td>373</td>
</tr>
<tr>
<td>459</td>
<td>374</td>
</tr>
<tr>
<td>460</td>
<td>366</td>
</tr>
<tr>
<td>461</td>
<td>388</td>
</tr>
<tr>
<td>462</td>
<td>379</td>
</tr>
<tr>
<td>463</td>
<td>378</td>
</tr>
<tr>
<td>464</td>
<td>370</td>
</tr>
<tr>
<td>465</td>
<td>374</td>
</tr>
<tr>
<td>466</td>
<td>366</td>
</tr>
<tr>
<td>467</td>
<td>373</td>
</tr>
<tr>
<td>468</td>
<td>381</td>
</tr>
<tr>
<td>469</td>
<td>379</td>
</tr>
<tr>
<td>470</td>
<td>379</td>
</tr>
<tr>
<td>471</td>
<td>375</td>
</tr>
<tr>
<td>472</td>
<td>370</td>
</tr>
<tr>
<td>473</td>
<td>380</td>
</tr>
<tr>
<td>474</td>
<td>359</td>
</tr>
<tr>
<td>475</td>
<td>372</td>
</tr>
<tr>
<td>476</td>
<td>370</td>
</tr>
<tr>
<td>477</td>
<td>368</td>
</tr>
<tr>
<td>478</td>
<td>385</td>
</tr>
<tr>
<td>479</td>
<td>370</td>
</tr>
<tr>
<td>480</td>
<td>382</td>
</tr>
<tr>
<td>481</td>
<td>378</td>
</tr>
<tr>
<td>482</td>
<td>388</td>
</tr>
<tr>
<td>483</td>
<td>380</td>
</tr>
<tr>
<td>484</td>
<td>384</td>
</tr>
<tr>
<td>485</td>
<td>384</td>
</tr>
<tr>
<td>486</td>
<td>384</td>
</tr>
<tr>
<td>487</td>
<td>376</td>
</tr>
<tr>
<td>488</td>
<td>383</td>
</tr>
<tr>
<td>489</td>
<td>368</td>
</tr>
<tr>
<td>490</td>
<td>375</td>
</tr>
<tr>
<td>491</td>
<td>381</td>
</tr>
<tr>
<td>492</td>
<td>373</td>
</tr>
<tr>
<td>493</td>
<td>377</td>
</tr>
<tr>
<td>494</td>
<td>379</td>
</tr>
<tr>
<td>495</td>
<td>369</td>
</tr>
<tr>
<td>496</td>
<td>333</td>
</tr>
<tr>
<td>497</td>
<td>386</td>
</tr>
<tr>
<td>498</td>
<td>374</td>
</tr>
<tr>
<td>499</td>
<td>385</td>
</tr>
<tr>
<td>500</td>
<td>374</td>
</tr>
<tr>
<td>Grand Total</td>
<td>18848</td>
</tr>
</tbody>
</table>
The seed number is 501 The total is 60
The seed number is 502 The total is 51
The seed number is 503 The total is 55
The seed number is 504 The total is 62
The seed number is 505 The total is 61
The seed number is 506 The total is 54
The seed number is 507 The total is 50
The seed number is 508 The total is 59
The seed number is 509 The total is 59
The seed number is 510 The total is 62
The seed number is 511 The total is 61
The seed number is 512 The total is 58
The seed number is 513 The total is 57
The seed number is 514 The total is 53
The seed number is 515 The total is 60
The seed number is 516 The total is 55
The seed number is 517 The total is 60
The seed number is 518 The total is 59
The seed number is 519 The total is 55
The seed number is 520 The total is 57
The seed number is 521 The total is 54
The seed number is 522 The total is 53
The seed number is 523 The total is 56
The seed number is 524 The total is 56
The seed number is 525 The total is 56
The seed number is 526 The total is 55
The seed number is 527 The total is 55
The seed number is 528 The total is 61
The seed number is 529 The total is 55
The seed number is 530 The total is 60
The seed number is 531 The total is 61
The seed number is 532 The total is 61
The seed number is 533 The total is 53
The seed number is 534 The total is 58
The seed number is 535 The total is 55
The seed number is 536 The total is 57
The seed number is 537 The total is 55
The seed number is 538 The total is 55
The seed number is 539 The total is 55
The seed number is 540 The total is 54
The seed number is 541 The total is 57
The seed number is 542 The total is 61
The seed number is 543 The total is 54
The seed number is 544 The total is 60
The seed number is 545 The total is 55
The seed number is 546 The total is 60
The seed number is 547 The total is 55
The seed number is 548 The total is 58
The seed number is 549 The total is 62
The seed number is 550 The total is 59

The grand total for all trials is 2851

The seed number is 551 The total is 54
The seed number is 552 The total is 57
The seed number is 553 The total is 54
The seed number is 554 The total is 59
The seed number is 555 The total is 59
The seed number is 556 The total is 59
The seed number is 557 The total is 54
The seed number is 558 The total is 59
The seed number is 559 The total is 57
The seed number is 560 The total is 59
The seed number is 561 The total is 49
The seed number is 562 The total is 57
The seed number is 563 The total is 60
The seed number is 564 The total is 59
The seed number is 565 The total is 57
The seed number is 566 The total is 55
The seed number is 567 The total is 54
The seed number is 568 The total is 59
The seed number is 569 The total is 55
The seed number is 570 The total is 61
The seed number is 571 The total is 57
The seed number is 572 The total is 54
The seed number is 573 The total is 59
The seed number is 574 The total is 55
The seed number is 575 The total is 51
The seed number is 576 The total is 61
The seed number is 577 The total is 52
The seed number is 578 The total is 53
The seed number is 579 The total is 57
The seed number is 580 The total is 59
The seed number is 581 The total is 60
The seed number is 582 The total is 52
The seed number is 583 The total is 55
The seed number is 584 The total is 60
The seed number is 585 The total is 52
The seed number is 586 The total is 58
The seed number is 587 The total is 62
The seed number is 588 The total is 55
The seed number is 589 The total is 61
The seed number is 590 The total is 63
The seed number is 591 The total is 58
The seed number is 592 The total is 52
The seed number is 593 The total is 56
The seed number is 594 The total is 61
The seed number is 595 The total is 55
The seed number is 596 The total is 53
The seed number is 597 The total is 58
The seed number is 598 The total is 58
The seed number is 599 The total is 56
The seed number is 600 The total is 53
The grand total for all trials is 2837
Results with the Hungarian Method

<table>
<thead>
<tr>
<th>hungarian method</th>
<th>hungarian method</th>
</tr>
</thead>
<tbody>
<tr>
<td>size is 13 range is 13</td>
<td>size is 13 range is 13</td>
</tr>
<tr>
<td>The seed is 1 The total is 154</td>
<td>The seed is 51 The total is 155</td>
</tr>
<tr>
<td>The seed is 2 The total is 157</td>
<td>The seed is 52 The total is 157</td>
</tr>
<tr>
<td>The seed is 3 The total is 155</td>
<td>The seed is 53 The total is 146</td>
</tr>
<tr>
<td>The seed is 4 The total is 150</td>
<td>The seed is 54 The total is 156</td>
</tr>
<tr>
<td>The seed is 5 The total is 163</td>
<td>The seed is 55 The total is 152</td>
</tr>
<tr>
<td>The seed is 6 The total is 159</td>
<td>The seed is 56 The total is 160</td>
</tr>
<tr>
<td>The seed is 7 The total is 156</td>
<td>The seed is 57 The total is 156</td>
</tr>
<tr>
<td>The seed is 8 The total is 154</td>
<td>The seed is 58 The total is 158</td>
</tr>
<tr>
<td>The seed is 9 The total is 158</td>
<td>The seed is 59 The total is 154</td>
</tr>
<tr>
<td>The seed is 10 The total is 157</td>
<td>The seed is 60 The total is 149</td>
</tr>
<tr>
<td>The seed is 11 The total is 156</td>
<td>The seed is 61 The total is 157</td>
</tr>
<tr>
<td>The seed is 12 The total is 153</td>
<td>The seed is 62 The total is 157</td>
</tr>
<tr>
<td>The seed is 13 The total is 150</td>
<td>The seed is 63 The total is 159</td>
</tr>
<tr>
<td>The seed is 14 The total is 165</td>
<td>The seed is 64 The total is 157</td>
</tr>
<tr>
<td>The seed is 15 The total is 158</td>
<td>The seed is 65 The total is 151</td>
</tr>
<tr>
<td>The seed is 16 The total is 157</td>
<td>The seed is 66 The total is 162</td>
</tr>
<tr>
<td>The seed is 17 The total is 159</td>
<td>The seed is 67 The total is 154</td>
</tr>
<tr>
<td>The seed is 18 The total is 157</td>
<td>The seed is 68 The total is 157</td>
</tr>
<tr>
<td>The seed is 19 The total is 151</td>
<td>The seed is 69 The total is 158</td>
</tr>
<tr>
<td>The seed is 20 The total is 161</td>
<td>The seed is 70 The total is 154</td>
</tr>
<tr>
<td>The seed is 21 The total is 153</td>
<td>The seed is 71 The total is 160</td>
</tr>
<tr>
<td>The seed is 22 The total is 156</td>
<td>The seed is 72 The total is 152</td>
</tr>
<tr>
<td>The seed is 23 The total is 156</td>
<td>The seed is 73 The total is 164</td>
</tr>
<tr>
<td>The seed is 24 The total is 154</td>
<td>The seed is 74 The total is 160</td>
</tr>
<tr>
<td>The seed is 25 The total is 157</td>
<td>The seed is 75 The total is 155</td>
</tr>
<tr>
<td>The seed is 26 The total is 152</td>
<td>The seed is 76 The total is 158</td>
</tr>
<tr>
<td>The seed is 27 The total is 157</td>
<td>The seed is 77 The total is 163</td>
</tr>
<tr>
<td>The seed is 28 The total is 160</td>
<td>The seed is 78 The total is 152</td>
</tr>
<tr>
<td>The seed is 29 The total is 158</td>
<td>The seed is 79 The total is 152</td>
</tr>
<tr>
<td>The seed is 30 The total is 160</td>
<td>The seed is 80 The total is 163</td>
</tr>
<tr>
<td>The seed is 31 The total is 159</td>
<td>The seed is 81 The total is 158</td>
</tr>
<tr>
<td>The seed is 32 The total is 161</td>
<td>The seed is 82 The total is 156</td>
</tr>
<tr>
<td>The seed is 33 The total is 145</td>
<td>The seed is 83 The total is 157</td>
</tr>
<tr>
<td>The seed is 34 The total is 153</td>
<td>The seed is 84 The total is 152</td>
</tr>
<tr>
<td>The seed is 35 The total is 152</td>
<td>The seed is 85 The total is 157</td>
</tr>
<tr>
<td>The seed is 36 The total is 158</td>
<td>The seed is 86 The total is 157</td>
</tr>
<tr>
<td>The seed is 37 The total is 152</td>
<td>The seed is 87 The total is 156</td>
</tr>
<tr>
<td>The seed is 38 The total is 157</td>
<td>The seed is 88 The total is 156</td>
</tr>
<tr>
<td>The seed is 39 The total is 160</td>
<td>The seed is 89 The total is 150</td>
</tr>
<tr>
<td>The seed is 40 The total is 153</td>
<td>The seed is 90 The total is 157</td>
</tr>
<tr>
<td>The seed is 41 The total is 153</td>
<td>The seed is 91 The total is 154</td>
</tr>
<tr>
<td>The seed is 42 The total is 149</td>
<td>The seed is 92 The total is 160</td>
</tr>
<tr>
<td>The seed is 43 The total is 159</td>
<td>The seed is 93 The total is 160</td>
</tr>
<tr>
<td>The seed is 44 The total is 158</td>
<td>The seed is 94 The total is 161</td>
</tr>
<tr>
<td>The seed is 45 The total is 137</td>
<td>The seed is 95 The total is 156</td>
</tr>
<tr>
<td>The seed is 46 The total is 156</td>
<td>The seed is 96 The total is 154</td>
</tr>
<tr>
<td>The seed is 47 The total is 160</td>
<td>The seed is 97 The total is 157</td>
</tr>
<tr>
<td>The seed is 48 The total is 157</td>
<td>The seed is 98 The total is 158</td>
</tr>
<tr>
<td>The seed is 49 The total is 153</td>
<td>The seed is 99 The total is 155</td>
</tr>
<tr>
<td>The seed is 50 The total is 156</td>
<td>The seed is 100 The total is 155</td>
</tr>
<tr>
<td>The grand total for all trials is 7801</td>
<td>The grand total for all trials is 7814</td>
</tr>
<tr>
<td>hungarian method</td>
<td>hungarian method</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------</td>
</tr>
<tr>
<td><strong>size is 13 range is 13</strong></td>
<td><strong>size is 13 range is 26</strong></td>
</tr>
<tr>
<td>The seed is 101 The total is 154</td>
<td>The seed is 151 The total is 304</td>
</tr>
<tr>
<td>The seed is 102 The total is 159</td>
<td>The seed is 152 The total is 311</td>
</tr>
<tr>
<td>The seed is 103 The total is 155</td>
<td>The seed is 153 The total is 320</td>
</tr>
<tr>
<td>The seed is 104 The total is 151</td>
<td>The seed is 154 The total is 309</td>
</tr>
<tr>
<td>The seed is 105 The total is 159</td>
<td>The seed is 155 The total is 318</td>
</tr>
<tr>
<td>The seed is 106 The total is 159</td>
<td>The seed is 156 The total is 303</td>
</tr>
<tr>
<td>The seed is 107 The total is 156</td>
<td>The seed is 157 The total is 300</td>
</tr>
<tr>
<td>The seed is 108 The total is 161</td>
<td>The seed is 158 The total is 311</td>
</tr>
<tr>
<td>The seed is 109 The total is 145</td>
<td>The seed is 159 The total is 303</td>
</tr>
<tr>
<td>The seed is 110 The total is 156</td>
<td>The seed is 160 The total is 304</td>
</tr>
<tr>
<td>The seed is 111 The total is 156</td>
<td>The seed is 161 The total is 307</td>
</tr>
<tr>
<td>The seed is 112 The total is 157</td>
<td>The seed is 162 The total is 295</td>
</tr>
<tr>
<td>The seed is 113 The total is 160</td>
<td>The seed is 163 The total is 299</td>
</tr>
<tr>
<td>The seed is 114 The total is 155</td>
<td>The seed is 164 The total is 305</td>
</tr>
<tr>
<td>The seed is 115 The total is 144</td>
<td>The seed is 165 The total is 310</td>
</tr>
<tr>
<td>The seed is 116 The total is 156</td>
<td>The seed is 166 The total is 310</td>
</tr>
<tr>
<td>The seed is 117 The total is 157</td>
<td>The seed is 167 The total is 304</td>
</tr>
<tr>
<td>The seed is 118 The total is 158</td>
<td>The seed is 168 The total is 308</td>
</tr>
<tr>
<td>The seed is 119 The total is 164</td>
<td>The seed is 169 The total is 288</td>
</tr>
<tr>
<td>The seed is 120 The total is 155</td>
<td>The seed is 170 The total is 308</td>
</tr>
<tr>
<td>The seed is 121 The total is 157</td>
<td>The seed is 171 The total is 300</td>
</tr>
<tr>
<td>The seed is 122 The total is 157</td>
<td>The seed is 172 The total is 303</td>
</tr>
<tr>
<td>The seed is 123 The total is 158</td>
<td>The seed is 173 The total is 318</td>
</tr>
<tr>
<td>The seed is 124 The total is 162</td>
<td>The seed is 174 The total is 304</td>
</tr>
<tr>
<td>The seed is 125 The total is 159</td>
<td>The seed is 175 The total is 315</td>
</tr>
<tr>
<td>The seed is 126 The total is 162</td>
<td>The seed is 176 The total is 306</td>
</tr>
<tr>
<td>The seed is 127 The total is 152</td>
<td>The seed is 177 The total is 311</td>
</tr>
<tr>
<td>The seed is 128 The total is 156</td>
<td>The seed is 178 The total is 302</td>
</tr>
<tr>
<td>The seed is 129 The total is 161</td>
<td>The seed is 179 The total is 310</td>
</tr>
<tr>
<td>The seed is 130 The total is 155</td>
<td>The seed is 180 The total is 300</td>
</tr>
<tr>
<td>The seed is 131 The total is 157</td>
<td>The seed is 181 The total is 307</td>
</tr>
<tr>
<td>The seed is 132 The total is 150</td>
<td>The seed is 182 The total is 303</td>
</tr>
<tr>
<td>The seed is 133 The total is 156</td>
<td>The seed is 183 The total is 304</td>
</tr>
<tr>
<td>The seed is 134 The total is 150</td>
<td>The seed is 184 The total is 305</td>
</tr>
<tr>
<td>The seed is 135 The total is 153</td>
<td>The seed is 185 The total is 308</td>
</tr>
<tr>
<td>The seed is 136 The total is 156</td>
<td>The seed is 186 The total is 314</td>
</tr>
<tr>
<td>The seed is 137 The total is 163</td>
<td>The seed is 187 The total is 314</td>
</tr>
<tr>
<td>The seed is 138 The total is 158</td>
<td>The seed is 188 The total is 299</td>
</tr>
<tr>
<td>The seed is 139 The total is 161</td>
<td>The seed is 189 The total is 293</td>
</tr>
<tr>
<td>The seed is 140 The total is 152</td>
<td>The seed is 190 The total is 301</td>
</tr>
<tr>
<td>The seed is 141 The total is 153</td>
<td>The seed is 191 The total is 315</td>
</tr>
<tr>
<td>The seed is 142 The total is 155</td>
<td>The seed is 192 The total is 307</td>
</tr>
<tr>
<td>The seed is 143 The total is 154</td>
<td>The seed is 193 The total is 297</td>
</tr>
<tr>
<td>The seed is 144 The total is 161</td>
<td>The seed is 194 The total is 313</td>
</tr>
<tr>
<td>The seed is 145 The total is 158</td>
<td>The seed is 195 The total is 310</td>
</tr>
<tr>
<td>The seed is 146 The total is 157</td>
<td>The seed is 196 The total is 310</td>
</tr>
<tr>
<td>The seed is 147 The total is 158</td>
<td>The seed is 197 The total is 305</td>
</tr>
<tr>
<td>The seed is 148 The total is 158</td>
<td>The seed is 198 The total is 299</td>
</tr>
<tr>
<td>The seed is 149 The total is 154</td>
<td>The seed is 199 The total is 313</td>
</tr>
<tr>
<td>The seed is 150 The total is 152</td>
<td>The seed is 200 The total is 293</td>
</tr>
</tbody>
</table>

The grand total for all trials is **7812**

The grand total for all trials is **15296**
<table>
<thead>
<tr>
<th>hungarian method</th>
<th>hungarian method</th>
</tr>
</thead>
<tbody>
<tr>
<td>size is 13 range is 26</td>
<td>size is 13 range is 26</td>
</tr>
<tr>
<td>The seed is 201 The total is 309</td>
<td>The seed is 251 The total is 315</td>
</tr>
<tr>
<td>The seed is 202 The total is 310</td>
<td>The seed is 252 The total is 299</td>
</tr>
<tr>
<td>The seed is 203 The total is 299</td>
<td>The seed is 253 The total is 310</td>
</tr>
<tr>
<td>The seed is 204 The total is 287</td>
<td>The seed is 254 The total is 320</td>
</tr>
<tr>
<td>The seed is 205 The total is 313</td>
<td>The seed is 255 The total is 301</td>
</tr>
<tr>
<td>The seed is 206 The total is 310</td>
<td>The seed is 256 The total is 310</td>
</tr>
<tr>
<td>The seed is 207 The total is 296</td>
<td>The seed is 257 The total is 314</td>
</tr>
<tr>
<td>The seed is 208 The total is 306</td>
<td>The seed is 258 The total is 309</td>
</tr>
<tr>
<td>The seed is 209 The total is 316</td>
<td>The seed is 259 The total is 307</td>
</tr>
<tr>
<td>The seed is 210 The total is 298</td>
<td>The seed is 260 The total is 289</td>
</tr>
<tr>
<td>The seed is 211 The total is 312</td>
<td>The seed is 261 The total is 309</td>
</tr>
<tr>
<td>The seed is 212 The total is 311</td>
<td>The seed is 262 The total is 298</td>
</tr>
<tr>
<td>The seed is 213 The total is 301</td>
<td>The seed is 263 The total is 318</td>
</tr>
<tr>
<td>The seed is 214 The total is 303</td>
<td>The seed is 264 The total is 310</td>
</tr>
<tr>
<td>The seed is 215 The total is 309</td>
<td>The seed is 265 The total is 305</td>
</tr>
<tr>
<td>The seed is 216 The total is 315</td>
<td>The seed is 266 The total is 306</td>
</tr>
<tr>
<td>The seed is 217 The total is 306</td>
<td>The seed is 267 The total is 311</td>
</tr>
<tr>
<td>The seed is 218 The total is 311</td>
<td>The seed is 268 The total is 311</td>
</tr>
<tr>
<td>The seed is 219 The total is 314</td>
<td>The seed is 269 The total is 308</td>
</tr>
<tr>
<td>The seed is 220 The total is 312</td>
<td>The seed is 270 The total is 307</td>
</tr>
<tr>
<td>The seed is 221 The total is 310</td>
<td>The seed is 271 The total is 305</td>
</tr>
<tr>
<td>The seed is 222 The total is 320</td>
<td>The seed is 272 The total is 298</td>
</tr>
<tr>
<td>The seed is 223 The total is 301</td>
<td>The seed is 273 The total is 309</td>
</tr>
<tr>
<td>The seed is 224 The total is 318</td>
<td>The seed is 274 The total is 307</td>
</tr>
<tr>
<td>The seed is 225 The total is 310</td>
<td>The seed is 275 The total is 306</td>
</tr>
<tr>
<td>The seed is 226 The total is 312</td>
<td>The seed is 276 The total is 300</td>
</tr>
<tr>
<td>The seed is 227 The total is 303</td>
<td>The seed is 277 The total is 314</td>
</tr>
<tr>
<td>The seed is 228 The total is 325</td>
<td>The seed is 278 The total is 303</td>
</tr>
<tr>
<td>The seed is 229 The total is 315</td>
<td>The seed is 279 The total is 291</td>
</tr>
<tr>
<td>The seed is 230 The total is 313</td>
<td>The seed is 280 The total is 277</td>
</tr>
<tr>
<td>The seed is 231 The total is 292</td>
<td>The seed is 281 The total is 315</td>
</tr>
<tr>
<td>The seed is 232 The total is 316</td>
<td>The seed is 282 The total is 312</td>
</tr>
<tr>
<td>The seed is 233 The total is 304</td>
<td>The seed is 283 The total is 311</td>
</tr>
<tr>
<td>The seed is 234 The total is 307</td>
<td>The seed is 284 The total is 314</td>
</tr>
<tr>
<td>The seed is 235 The total is 293</td>
<td>The seed is 285 The total is 318</td>
</tr>
<tr>
<td>The seed is 236 The total is 306</td>
<td>The seed is 286 The total is 321</td>
</tr>
<tr>
<td>The seed is 237 The total is 307</td>
<td>The seed is 287 The total is 310</td>
</tr>
<tr>
<td>The seed is 238 The total is 317</td>
<td>The seed is 288 The total is 320</td>
</tr>
<tr>
<td>The seed is 239 The total is 310</td>
<td>The seed is 289 The total is 301</td>
</tr>
<tr>
<td>The seed is 240 The total is 319</td>
<td>The seed is 290 The total is 298</td>
</tr>
<tr>
<td>The seed is 241 The total is 299</td>
<td>The seed is 291 The total is 311</td>
</tr>
<tr>
<td>The seed is 242 The total is 317</td>
<td>The seed is 292 The total is 315</td>
</tr>
<tr>
<td>The seed is 243 The total is 305</td>
<td>The seed is 293 The total is 308</td>
</tr>
<tr>
<td>The seed is 244 The total is 299</td>
<td>The seed is 294 The total is 305</td>
</tr>
<tr>
<td>The seed is 245 The total is 297</td>
<td>The seed is 295 The total is 307</td>
</tr>
<tr>
<td>The seed is 246 The total is 310</td>
<td>The seed is 296 The total is 319</td>
</tr>
<tr>
<td>The seed is 247 The total is 317</td>
<td>The seed is 297 The total is 298</td>
</tr>
<tr>
<td>The seed is 248 The total is 300</td>
<td>The seed is 298 The total is 312</td>
</tr>
<tr>
<td>The seed is 249 The total is 309</td>
<td>The seed is 299 The total is 315</td>
</tr>
<tr>
<td>The seed is 250 The total is 312</td>
<td>The seed is 300 The total is 283</td>
</tr>
<tr>
<td>The grand total for all trials is 15401</td>
<td>The grand total for all trials is 15360</td>
</tr>
</tbody>
</table>
Hungarian method
Size is 20 range is 20

<table>
<thead>
<tr>
<th>Seed</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>401</td>
<td>380</td>
</tr>
<tr>
<td>402</td>
<td>382</td>
</tr>
<tr>
<td>403</td>
<td>377</td>
</tr>
<tr>
<td>404</td>
<td>376</td>
</tr>
<tr>
<td>405</td>
<td>383</td>
</tr>
<tr>
<td>406</td>
<td>384</td>
</tr>
<tr>
<td>407</td>
<td>375</td>
</tr>
<tr>
<td>408</td>
<td>377</td>
</tr>
<tr>
<td>409</td>
<td>371</td>
</tr>
<tr>
<td>410</td>
<td>380</td>
</tr>
<tr>
<td>411</td>
<td>379</td>
</tr>
<tr>
<td>412</td>
<td>392</td>
</tr>
<tr>
<td>413</td>
<td>381</td>
</tr>
<tr>
<td>414</td>
<td>375</td>
</tr>
<tr>
<td>415</td>
<td>375</td>
</tr>
<tr>
<td>416</td>
<td>375</td>
</tr>
<tr>
<td>417</td>
<td>385</td>
</tr>
<tr>
<td>418</td>
<td>379</td>
</tr>
<tr>
<td>419</td>
<td>385</td>
</tr>
<tr>
<td>420</td>
<td>375</td>
</tr>
<tr>
<td>421</td>
<td>381</td>
</tr>
<tr>
<td>422</td>
<td>379</td>
</tr>
<tr>
<td>423</td>
<td>383</td>
</tr>
<tr>
<td>424</td>
<td>379</td>
</tr>
<tr>
<td>425</td>
<td>379</td>
</tr>
<tr>
<td>426</td>
<td>386</td>
</tr>
<tr>
<td>427</td>
<td>378</td>
</tr>
<tr>
<td>428</td>
<td>384</td>
</tr>
<tr>
<td>429</td>
<td>381</td>
</tr>
<tr>
<td>430</td>
<td>385</td>
</tr>
<tr>
<td>431</td>
<td>366</td>
</tr>
<tr>
<td>432</td>
<td>379</td>
</tr>
<tr>
<td>433</td>
<td>375</td>
</tr>
<tr>
<td>434</td>
<td>381</td>
</tr>
<tr>
<td>435</td>
<td>381</td>
</tr>
<tr>
<td>436</td>
<td>388</td>
</tr>
<tr>
<td>437</td>
<td>380</td>
</tr>
<tr>
<td>438</td>
<td>381</td>
</tr>
<tr>
<td>439</td>
<td>383</td>
</tr>
<tr>
<td>440</td>
<td>375</td>
</tr>
<tr>
<td>441</td>
<td>387</td>
</tr>
<tr>
<td>442</td>
<td>379</td>
</tr>
<tr>
<td>443</td>
<td>386</td>
</tr>
<tr>
<td>444</td>
<td>386</td>
</tr>
<tr>
<td>445</td>
<td>382</td>
</tr>
<tr>
<td>446</td>
<td>384</td>
</tr>
<tr>
<td>447</td>
<td>385</td>
</tr>
<tr>
<td>448</td>
<td>383</td>
</tr>
<tr>
<td>449</td>
<td>377</td>
</tr>
<tr>
<td>450</td>
<td>381</td>
</tr>
<tr>
<td>451</td>
<td>381</td>
</tr>
<tr>
<td>452</td>
<td>379</td>
</tr>
<tr>
<td>453</td>
<td>381</td>
</tr>
<tr>
<td>454</td>
<td>387</td>
</tr>
<tr>
<td>455</td>
<td>378</td>
</tr>
<tr>
<td>456</td>
<td>382</td>
</tr>
<tr>
<td>457</td>
<td>385</td>
</tr>
<tr>
<td>458</td>
<td>378</td>
</tr>
<tr>
<td>459</td>
<td>381</td>
</tr>
<tr>
<td>460</td>
<td>376</td>
</tr>
<tr>
<td>461</td>
<td>388</td>
</tr>
<tr>
<td>462</td>
<td>381</td>
</tr>
<tr>
<td>463</td>
<td>382</td>
</tr>
<tr>
<td>464</td>
<td>376</td>
</tr>
<tr>
<td>465</td>
<td>379</td>
</tr>
<tr>
<td>466</td>
<td>372</td>
</tr>
<tr>
<td>467</td>
<td>376</td>
</tr>
<tr>
<td>468</td>
<td>381</td>
</tr>
<tr>
<td>469</td>
<td>380</td>
</tr>
<tr>
<td>470</td>
<td>383</td>
</tr>
<tr>
<td>471</td>
<td>386</td>
</tr>
<tr>
<td>472</td>
<td>378</td>
</tr>
<tr>
<td>473</td>
<td>385</td>
</tr>
<tr>
<td>474</td>
<td>374</td>
</tr>
<tr>
<td>475</td>
<td>378</td>
</tr>
<tr>
<td>476</td>
<td>376</td>
</tr>
<tr>
<td>477</td>
<td>379</td>
</tr>
<tr>
<td>478</td>
<td>385</td>
</tr>
<tr>
<td>479</td>
<td>379</td>
</tr>
<tr>
<td>480</td>
<td>382</td>
</tr>
<tr>
<td>481</td>
<td>378</td>
</tr>
<tr>
<td>482</td>
<td>388</td>
</tr>
<tr>
<td>483</td>
<td>383</td>
</tr>
<tr>
<td>484</td>
<td>386</td>
</tr>
<tr>
<td>485</td>
<td>381</td>
</tr>
<tr>
<td>486</td>
<td>385</td>
</tr>
<tr>
<td>487</td>
<td>382</td>
</tr>
<tr>
<td>488</td>
<td>383</td>
</tr>
<tr>
<td>489</td>
<td>373</td>
</tr>
<tr>
<td>490</td>
<td>375</td>
</tr>
<tr>
<td>491</td>
<td>384</td>
</tr>
<tr>
<td>492</td>
<td>376</td>
</tr>
<tr>
<td>493</td>
<td>380</td>
</tr>
<tr>
<td>494</td>
<td>384</td>
</tr>
<tr>
<td>495</td>
<td>384</td>
</tr>
<tr>
<td>496</td>
<td>383</td>
</tr>
<tr>
<td>497</td>
<td>387</td>
</tr>
<tr>
<td>498</td>
<td>378</td>
</tr>
<tr>
<td>499</td>
<td>386</td>
</tr>
<tr>
<td>500</td>
<td>379</td>
</tr>
</tbody>
</table>

Grand total for all trials is 19013

Hungarian method
Size is 20 range is 20

<table>
<thead>
<tr>
<th>Seed</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>451</td>
<td>381</td>
</tr>
<tr>
<td>452</td>
<td>379</td>
</tr>
<tr>
<td>453</td>
<td>381</td>
</tr>
<tr>
<td>454</td>
<td>387</td>
</tr>
<tr>
<td>455</td>
<td>378</td>
</tr>
<tr>
<td>456</td>
<td>382</td>
</tr>
<tr>
<td>457</td>
<td>385</td>
</tr>
<tr>
<td>458</td>
<td>378</td>
</tr>
<tr>
<td>459</td>
<td>381</td>
</tr>
<tr>
<td>460</td>
<td>376</td>
</tr>
<tr>
<td>461</td>
<td>388</td>
</tr>
<tr>
<td>462</td>
<td>381</td>
</tr>
<tr>
<td>463</td>
<td>382</td>
</tr>
<tr>
<td>464</td>
<td>376</td>
</tr>
<tr>
<td>465</td>
<td>379</td>
</tr>
<tr>
<td>466</td>
<td>372</td>
</tr>
<tr>
<td>467</td>
<td>376</td>
</tr>
<tr>
<td>468</td>
<td>381</td>
</tr>
<tr>
<td>469</td>
<td>380</td>
</tr>
<tr>
<td>470</td>
<td>383</td>
</tr>
<tr>
<td>471</td>
<td>386</td>
</tr>
<tr>
<td>472</td>
<td>378</td>
</tr>
<tr>
<td>473</td>
<td>385</td>
</tr>
<tr>
<td>474</td>
<td>374</td>
</tr>
<tr>
<td>475</td>
<td>378</td>
</tr>
<tr>
<td>476</td>
<td>376</td>
</tr>
<tr>
<td>477</td>
<td>379</td>
</tr>
<tr>
<td>478</td>
<td>385</td>
</tr>
<tr>
<td>479</td>
<td>379</td>
</tr>
<tr>
<td>480</td>
<td>382</td>
</tr>
<tr>
<td>481</td>
<td>378</td>
</tr>
<tr>
<td>482</td>
<td>388</td>
</tr>
<tr>
<td>483</td>
<td>383</td>
</tr>
<tr>
<td>484</td>
<td>386</td>
</tr>
<tr>
<td>485</td>
<td>381</td>
</tr>
<tr>
<td>486</td>
<td>385</td>
</tr>
<tr>
<td>487</td>
<td>382</td>
</tr>
<tr>
<td>488</td>
<td>383</td>
</tr>
<tr>
<td>489</td>
<td>373</td>
</tr>
<tr>
<td>490</td>
<td>375</td>
</tr>
<tr>
<td>491</td>
<td>384</td>
</tr>
<tr>
<td>492</td>
<td>376</td>
</tr>
<tr>
<td>493</td>
<td>380</td>
</tr>
<tr>
<td>494</td>
<td>384</td>
</tr>
<tr>
<td>495</td>
<td>384</td>
</tr>
<tr>
<td>496</td>
<td>383</td>
</tr>
<tr>
<td>497</td>
<td>387</td>
</tr>
<tr>
<td>498</td>
<td>378</td>
</tr>
<tr>
<td>499</td>
<td>386</td>
</tr>
<tr>
<td>500</td>
<td>379</td>
</tr>
</tbody>
</table>

Grand total for all trials is 19029

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
<table>
<thead>
<tr>
<th>hungarian method</th>
<th>hungarian method</th>
</tr>
</thead>
<tbody>
<tr>
<td>size is 8 range is 8</td>
<td>size is 8 range is 8</td>
</tr>
<tr>
<td>The seed is 501 The total is 60</td>
<td>The seed is 551 The total is 54</td>
</tr>
<tr>
<td>The seed is 502 The total is 53</td>
<td>The seed is 552 The total is 59</td>
</tr>
<tr>
<td>The seed is 503 The total is 55</td>
<td>The seed is 553 The total is 56</td>
</tr>
<tr>
<td>The seed is 504 The total is 62</td>
<td>The seed is 554 The total is 59</td>
</tr>
<tr>
<td>The seed is 505 The total is 61</td>
<td>The seed is 555 The total is 59</td>
</tr>
<tr>
<td>The seed is 506 The total is 56</td>
<td>The seed is 556 The total is 54</td>
</tr>
<tr>
<td>The seed is 507 The total is 53</td>
<td>The seed is 557 The total is 54</td>
</tr>
<tr>
<td>The seed is 508 The total is 59</td>
<td>The seed is 558 The total is 59</td>
</tr>
<tr>
<td>The seed is 509 The total is 60</td>
<td>The seed is 559 The total is 57</td>
</tr>
<tr>
<td>The seed is 510 The total is 62</td>
<td>The seed is 560 The total is 59</td>
</tr>
<tr>
<td>The seed is 511 The total is 61</td>
<td>The seed is 561 The total is 50</td>
</tr>
<tr>
<td>The seed is 512 The total is 58</td>
<td>The seed is 562 The total is 57</td>
</tr>
<tr>
<td>The seed is 513 The total is 58</td>
<td>The seed is 563 The total is 60</td>
</tr>
<tr>
<td>The seed is 514 The total is 54</td>
<td>The seed is 564 The total is 59</td>
</tr>
<tr>
<td>The seed is 515 The total is 60</td>
<td>The seed is 565 The total is 57</td>
</tr>
<tr>
<td>The seed is 516 The total is 53</td>
<td>The seed is 566 The total is 55</td>
</tr>
<tr>
<td>The seed is 517 The total is 60</td>
<td>The seed is 567 The total is 56</td>
</tr>
<tr>
<td>The seed is 518 The total is 59</td>
<td>The seed is 568 The total is 59</td>
</tr>
<tr>
<td>The seed is 519 The total is 55</td>
<td>The seed is 569 The total is 55</td>
</tr>
<tr>
<td>The seed is 520 The total is 57</td>
<td>The seed is 570 The total is 61</td>
</tr>
<tr>
<td>The seed is 521 The total is 55</td>
<td>The seed is 571 The total is 57</td>
</tr>
<tr>
<td>The seed is 522 The total is 53</td>
<td>The seed is 572 The total is 54</td>
</tr>
<tr>
<td>The seed is 523 The total is 57</td>
<td>The seed is 573 The total is 59</td>
</tr>
<tr>
<td>The seed is 524 The total is 57</td>
<td>The seed is 574 The total is 55</td>
</tr>
<tr>
<td>The seed is 525 The total is 57</td>
<td>The seed is 575 The total is 51</td>
</tr>
<tr>
<td>The seed is 526 The total is 55</td>
<td>The seed is 576 The total is 61</td>
</tr>
<tr>
<td>The seed is 527 The total is 55</td>
<td>The seed is 577 The total is 53</td>
</tr>
<tr>
<td>The seed is 528 The total is 61</td>
<td>The seed is 578 The total is 55</td>
</tr>
<tr>
<td>The seed is 529 The total is 55</td>
<td>The seed is 579 The total is 57</td>
</tr>
<tr>
<td>The seed is 530 The total is 60</td>
<td>The seed is 580 The total is 60</td>
</tr>
<tr>
<td>The seed is 531 The total is 61</td>
<td>The seed is 581 The total is 60</td>
</tr>
<tr>
<td>The seed is 532 The total is 61</td>
<td>The seed is 582 The total is 55</td>
</tr>
<tr>
<td>The seed is 533 The total is 56</td>
<td>The seed is 583 The total is 55</td>
</tr>
<tr>
<td>The seed is 534 The total is 58</td>
<td>The seed is 584 The total is 60</td>
</tr>
<tr>
<td>The seed is 535 The total is 54</td>
<td>The seed is 585 The total is 61</td>
</tr>
<tr>
<td>The seed is 536 The total is 58</td>
<td>The seed is 586 The total is 58</td>
</tr>
<tr>
<td>The seed is 537 The total is 55</td>
<td>The seed is 587 The total is 62</td>
</tr>
<tr>
<td>The seed is 538 The total is 55</td>
<td>The seed is 588 The total is 55</td>
</tr>
<tr>
<td>The seed is 539 The total is 55</td>
<td>The seed is 589 The total is 59</td>
</tr>
<tr>
<td>The seed is 540 The total is 56</td>
<td>The seed is 590 The total is 63</td>
</tr>
<tr>
<td>The seed is 541 The total is 60</td>
<td>The seed is 591 The total is 58</td>
</tr>
<tr>
<td>The seed is 542 The total is 62</td>
<td>The seed is 592 The total is 53</td>
</tr>
<tr>
<td>The seed is 543 The total is 55</td>
<td>The seed is 593 The total is 56</td>
</tr>
<tr>
<td>The seed is 544 The total is 60</td>
<td>The seed is 594 The total is 61</td>
</tr>
<tr>
<td>The seed is 545 The total is 56</td>
<td>The seed is 595 The total is 55</td>
</tr>
<tr>
<td>The seed is 546 The total is 60</td>
<td>The seed is 596 The total is 54</td>
</tr>
<tr>
<td>The seed is 547 The total is 55</td>
<td>The seed is 597 The total is 58</td>
</tr>
<tr>
<td>The seed is 548 The total is 58</td>
<td>The seed is 598 The total is 58</td>
</tr>
<tr>
<td>The seed is 549 The total is 62</td>
<td>The seed is 599 The total is 57</td>
</tr>
<tr>
<td>The seed is 550 The total is 60</td>
<td>The seed is 600 The total is 54</td>
</tr>
<tr>
<td>The grand total for all trials is 2880</td>
<td>The grand total for all trials is 2855</td>
</tr>
</tbody>
</table>
Results with the Neural Net Method

<table>
<thead>
<tr>
<th>Neural Net Method</th>
<th>Neural Net Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>the size is 10</td>
<td>the size is 10</td>
</tr>
<tr>
<td>the multiplier is 2</td>
<td>the multiplier is 2</td>
</tr>
<tr>
<td>The seed is 1</td>
<td>The seed is 1</td>
</tr>
<tr>
<td>The total is 88</td>
<td>The total is 88</td>
</tr>
<tr>
<td>The seed is 2</td>
<td>The seed is 2</td>
</tr>
<tr>
<td>The total is 92</td>
<td>The total is 92</td>
</tr>
<tr>
<td>The seed is 3</td>
<td>The seed is 3</td>
</tr>
<tr>
<td>The total is 93</td>
<td>The total is 93</td>
</tr>
<tr>
<td>The seed is 4</td>
<td>The seed is 4</td>
</tr>
<tr>
<td>The total is 86</td>
<td>The total is 86</td>
</tr>
<tr>
<td>The seed is 5</td>
<td>The seed is 5</td>
</tr>
<tr>
<td>The total is 90</td>
<td>The total is 90</td>
</tr>
<tr>
<td>The seed is 6</td>
<td>The seed is 6</td>
</tr>
<tr>
<td>The total is 94</td>
<td>The total is 94</td>
</tr>
<tr>
<td>The seed is 7</td>
<td>The seed is 7</td>
</tr>
<tr>
<td>The total is 93</td>
<td>The total is 93</td>
</tr>
<tr>
<td>The seed is 8</td>
<td>The seed is 8</td>
</tr>
<tr>
<td>The total is 83</td>
<td>The total is 83</td>
</tr>
<tr>
<td>The seed is 9</td>
<td>The seed is 9</td>
</tr>
<tr>
<td>The total is 91</td>
<td>The total is 91</td>
</tr>
<tr>
<td>The seed is 10</td>
<td>The seed is 10</td>
</tr>
<tr>
<td>The total is 90</td>
<td>The total is 90</td>
</tr>
<tr>
<td>The seed is 11</td>
<td>The seed is 11</td>
</tr>
<tr>
<td>The total is 88</td>
<td>The total is 88</td>
</tr>
<tr>
<td>The seed is 12</td>
<td>The seed is 12</td>
</tr>
<tr>
<td>The total is 92</td>
<td>The total is 92</td>
</tr>
<tr>
<td>The seed is 13</td>
<td>The seed is 13</td>
</tr>
<tr>
<td>The total is 92</td>
<td>The total is 92</td>
</tr>
<tr>
<td>The seed is 14</td>
<td>The seed is 14</td>
</tr>
<tr>
<td>The total is 94</td>
<td>The total is 94</td>
</tr>
<tr>
<td>The seed is 15</td>
<td>The seed is 15</td>
</tr>
<tr>
<td>The total is 94</td>
<td>The total is 94</td>
</tr>
<tr>
<td>The seed is 16</td>
<td>The seed is 16</td>
</tr>
<tr>
<td>The total is 91</td>
<td>The total is 91</td>
</tr>
<tr>
<td>The seed is 17</td>
<td>The seed is 17</td>
</tr>
<tr>
<td>The total is 89</td>
<td>The total is 89</td>
</tr>
<tr>
<td>The seed is 18</td>
<td>The seed is 18</td>
</tr>
<tr>
<td>The total is 91</td>
<td>The total is 91</td>
</tr>
<tr>
<td>The seed is 19</td>
<td>The seed is 19</td>
</tr>
<tr>
<td>The total is 96</td>
<td>The total is 96</td>
</tr>
<tr>
<td>The seed is 20</td>
<td>The seed is 20</td>
</tr>
<tr>
<td>The total is 94</td>
<td>The total is 94</td>
</tr>
<tr>
<td>The seed is 21</td>
<td>The seed is 21</td>
</tr>
<tr>
<td>The total is 92</td>
<td>The total is 92</td>
</tr>
<tr>
<td>The seed is 22</td>
<td>The seed is 22</td>
</tr>
<tr>
<td>The total is 92</td>
<td>The total is 92</td>
</tr>
<tr>
<td>The seed is 23</td>
<td>The seed is 23</td>
</tr>
<tr>
<td>The total is 82</td>
<td>The total is 82</td>
</tr>
<tr>
<td>The seed is 24</td>
<td>The seed is 24</td>
</tr>
<tr>
<td>The total is 88</td>
<td>The total is 88</td>
</tr>
<tr>
<td>The seed is 25</td>
<td>The seed is 25</td>
</tr>
<tr>
<td>The total is 84</td>
<td>The total is 84</td>
</tr>
<tr>
<td>The seed is 26</td>
<td>The seed is 26</td>
</tr>
<tr>
<td>The total is 94</td>
<td>The total is 94</td>
</tr>
<tr>
<td>The seed is 27</td>
<td>The seed is 27</td>
</tr>
<tr>
<td>The total is 94</td>
<td>The total is 94</td>
</tr>
<tr>
<td>The seed is 28</td>
<td>The seed is 28</td>
</tr>
<tr>
<td>The total is 88</td>
<td>The total is 88</td>
</tr>
<tr>
<td>The seed is 29</td>
<td>The seed is 29</td>
</tr>
<tr>
<td>The total is 88</td>
<td>The total is 88</td>
</tr>
<tr>
<td>The seed is 30</td>
<td>The seed is 30</td>
</tr>
<tr>
<td>The total is 85</td>
<td>The total is 85</td>
</tr>
<tr>
<td>The seed is 31</td>
<td>The seed is 31</td>
</tr>
<tr>
<td>The total is 92</td>
<td>The total is 92</td>
</tr>
<tr>
<td>The seed is 32</td>
<td>The seed is 32</td>
</tr>
<tr>
<td>The total is 94</td>
<td>The total is 94</td>
</tr>
<tr>
<td>The seed is 33</td>
<td>The seed is 33</td>
</tr>
<tr>
<td>The total is 92</td>
<td>The total is 92</td>
</tr>
<tr>
<td>The seed is 34</td>
<td>The seed is 34</td>
</tr>
<tr>
<td>The total is 90</td>
<td>The total is 90</td>
</tr>
<tr>
<td>The seed is 35</td>
<td>The seed is 35</td>
</tr>
<tr>
<td>The total is 95</td>
<td>The total is 95</td>
</tr>
<tr>
<td>The seed is 36</td>
<td>The seed is 36</td>
</tr>
<tr>
<td>The total is 89</td>
<td>The total is 89</td>
</tr>
<tr>
<td>The seed is 37</td>
<td>The seed is 37</td>
</tr>
<tr>
<td>The total is 90</td>
<td>The total is 90</td>
</tr>
<tr>
<td>The seed is 38</td>
<td>The seed is 38</td>
</tr>
<tr>
<td>The total is 97</td>
<td>The total is 97</td>
</tr>
<tr>
<td>The seed is 39</td>
<td>The seed is 39</td>
</tr>
<tr>
<td>The total is 93</td>
<td>The total is 93</td>
</tr>
<tr>
<td>The seed is 40</td>
<td>The seed is 40</td>
</tr>
<tr>
<td>The total is 93</td>
<td>The total is 93</td>
</tr>
<tr>
<td>The seed is 41</td>
<td>The seed is 41</td>
</tr>
<tr>
<td>The total is 97</td>
<td>The total is 97</td>
</tr>
<tr>
<td>The seed is 42</td>
<td>The seed is 42</td>
</tr>
<tr>
<td>The total is 89</td>
<td>The total is 89</td>
</tr>
<tr>
<td>The seed is 43</td>
<td>The seed is 43</td>
</tr>
<tr>
<td>The total is 87</td>
<td>The total is 87</td>
</tr>
<tr>
<td>The seed is 44</td>
<td>The seed is 44</td>
</tr>
<tr>
<td>The total is 95</td>
<td>The total is 95</td>
</tr>
<tr>
<td>The seed is 45</td>
<td>The seed is 45</td>
</tr>
<tr>
<td>The total is 90</td>
<td>The total is 90</td>
</tr>
<tr>
<td>The seed is 46</td>
<td>The seed is 46</td>
</tr>
<tr>
<td>The total is 95</td>
<td>The total is 95</td>
</tr>
<tr>
<td>The seed is 47</td>
<td>The seed is 47</td>
</tr>
<tr>
<td>The total is 88</td>
<td>The total is 88</td>
</tr>
<tr>
<td>The seed is 48</td>
<td>The seed is 48</td>
</tr>
<tr>
<td>The total is 94</td>
<td>The total is 94</td>
</tr>
<tr>
<td>The seed is 49</td>
<td>The seed is 49</td>
</tr>
<tr>
<td>The total is 89</td>
<td>The total is 89</td>
</tr>
<tr>
<td>The seed is 50</td>
<td>The seed is 50</td>
</tr>
<tr>
<td>The total is 93</td>
<td>The total is 93</td>
</tr>
<tr>
<td>The grand total for all trials is 4550</td>
<td>The grand total for all trials is 4495</td>
</tr>
</tbody>
</table>

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
neural net approach known maximum
the size is 10 the multiplier is 2
the range is 10
The seed is 1 The total is 100
The seed is 2 The total is 100
The seed is 3 The total is 100
The seed is 4 The total is 100
The seed is 5 The total is 100
The seed is 6 The total is 100
The seed is 7 The total is 100
The seed is 8 The total is 100
The seed is 9 The total is 100
The seed is 10 The total is 100
The seed is 11 The total is 100
The seed is 12 The total is 100
The seed is 13 The total is 100
The seed is 14 The total is 100
The seed is 15 The total is 100
The seed is 16 The total is 100
The seed is 17 The total is 100
The seed is 18 The total is 100
The seed is 19 The total is 100
The seed is 20 The total is 100
The seed is 21 The total is 100
The seed is 22 The total is 100
The seed is 23 The total is 100
The seed is 24 The total is 100
The seed is 25 The total is 100
The seed is 26 The total is 100
The seed is 27 The total is 100
The seed is 28 The total is 99
The seed is 29 The total is 100
The seed is 30 The total is 100
The seed is 31 The total is 100
The seed is 32 The total is 100
The seed is 33 The total is 100
The seed is 34 The total is 100
The seed is 35 The total is 98
The seed is 36 The total is 100
The seed is 37 The total is 100
The seed is 38 The total is 100
The seed is 39 The total is 100
The seed is 40 The total is 100
The seed is 41 The total is 100
The seed is 42 The total is 100
The seed is 43 The total is 100
The seed is 44 The total is 100
The seed is 45 The total is 100
The seed is 46 The total is 99
The seed is 47 The total is 100
The seed is 48 The total is 100
The seed is 49 The total is 100
The seed is 50 The total is 100
The grand total for all trials is 4996

neural net approach known maximum
the size is 10 the multiplier is 3
the range is 10
The seed is 1 The total is 100
The seed is 2 The total is 100
The seed is 3 The total is 100
The seed is 4 The total is 100
The seed is 5 The total is 100
The seed is 6 The total is 100
The seed is 7 The total is 100
The seed is 8 The total is 100
The seed is 9 The total is 100
The seed is 10 The total is 100
The seed is 11 The total is 100
The seed is 12 The total is 100
The seed is 13 The total is 100
The seed is 14 The total is 100
The seed is 15 The total is 100
The seed is 16 The total is 100
The seed is 17 The total is 100
The seed is 18 The total is 100
The seed is 19 The total is 100
The seed is 20 The total is 100
The seed is 21 The total is 100
The seed is 22 The total is 100
The seed is 23 The total is 100
The seed is 24 The total is 100
The seed is 25 The total is 100
The seed is 26 The total is 100
The seed is 27 The total is 100
The seed is 28 The total is 100
The seed is 29 The total is 100
The seed is 30 The total is 100
The seed is 31 The total is 100
The seed is 32 The total is 100
The seed is 33 The total is 100
The seed is 34 The total is 100
The seed is 35 The total is 99
The seed is 36 The total is 100
The seed is 37 The total is 100
The seed is 38 The total is 100
The seed is 39 The total is 100
The seed is 40 The total is 100
The seed is 41 The total is 100
The seed is 42 The total is 100
The seed is 43 The total is 100
The seed is 44 The total is 100
The seed is 45 The total is 100
The seed is 46 The total is 99
The seed is 47 The total is 100
The seed is 48 The total is 100
The seed is 49 The total is 100
The seed is 50 The total is 100
The grand total for all trials is 4992

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
neural net approach known maximum
the size is 10 the multiplier is 5
the range is 10
The seed is 1 The total is 100
The seed is 2 The total is 100
The seed is 3 The total is 100
The seed is 4 The total is 100
The seed is 5 The total is 100
The seed is 6 The total is 100
The seed is 7 The total is 100
The seed is 8 The total is 100
The seed is 9 The total is 100
The seed is 10 The total is 100
The seed is 11 The total is 100
The seed is 12 The total is 100
The seed is 13 The total is 100
The seed is 14 The total is 100
The seed is 15 The total is 100
The seed is 16 The total is 100
The seed is 17 The total is 100
The seed is 18 The total is 100
The seed is 19 The total is 99
The seed is 20 The total is 100
The seed is 21 The total is 100
The seed is 22 The total is 100
The seed is 23 The total is 100
The seed is 24 The total is 100
The seed is 25 The total is 100
The seed is 26 The total is 100
The seed is 27 The total is 100
The seed is 28 The total is 100
The seed is 29 The total is 100
The seed is 30 The total is 100
The seed is 31 The total is 100
The seed is 32 The total is 100
The seed is 33 The total is 100
The seed is 34 The total is 100
The seed is 35 The total is 100
The seed is 36 The total is 100
The seed is 37 The total is 100
The seed is 38 The total is 100
The seed is 39 The total is 100
The seed is 40 The total is 97
The seed is 41 The total is 100
The seed is 42 The total is 100
The seed is 43 The total is 99
The seed is 44 The total is 100
The seed is 45 The total is 100
The seed is 46 The total is 99
The seed is 47 The total is 100
The seed is 48 The total is 100
The seed is 49 The total is 100
The seed is 50 The total is 100
The grand total for all trials is 4992

neural net approach known maximum
the size is 10 the multiplier is 10
the range is 10
The seed is 1 The total is 100
The seed is 2 The total is 100
The seed is 3 The total is 100
The seed is 4 The total is 100
The seed is 5 The total is 100
The seed is 6 The total is 100
The seed is 7 The total is 100
The seed is 8 The total is 100
The seed is 9 The total is 100
The seed is 10 The total is 100
The seed is 11 The total is 100
The seed is 12 The total is 100
The seed is 13 The total is 100
The seed is 14 The total is 100
The seed is 15 The total is 100
The seed is 16 The total is 100
The seed is 17 The total is 100
The seed is 18 The total is 100
The seed is 19 The total is 99
The seed is 20 The total is 100
The seed is 21 The total is 100
The seed is 22 The total is 100
The seed is 23 The total is 100
The seed is 24 The total is 100
The seed is 25 The total is 100
The seed is 26 The total is 100
The seed is 27 The total is 100
The seed is 28 The total is 100
The seed is 29 The total is 100
The seed is 30 The total is 100
The seed is 31 The total is 100
The seed is 32 The total is 100
The seed is 33 The total is 100
The seed is 34 The total is 100
The seed is 35 The total is 100
The seed is 36 The total is 100
The seed is 37 The total is 100
The seed is 38 The total is 100
The seed is 39 The total is 100
The seed is 40 The total is 97
The seed is 41 The total is 100
The seed is 42 The total is 100
The seed is 43 The total is 99
The seed is 44 The total is 100
The seed is 45 The total is 100
The seed is 46 The total is 99
The seed is 47 The total is 100
The seed is 48 The total is 100
The seed is 49 The total is 100
The seed is 50 The total is 100
The grand total for all trials is 4992

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
neural net approach known maximum
the size is 10 the multiplier is 2
the range is 10
The seed is 1 The total is 100
The seed is 2 The total is 100
The seed is 3 The total is 100
The seed is 4 The total is 100
The seed is 5 The total is 100
The seed is 6 The total is 100
The seed is 7 The total is 100
The seed is 8 The total is 100
The seed is 9 The total is 100
The seed is 10 The total is 100
The seed is 11 The total is 100
The seed is 12 The total is 100
The seed is 13 The total is 100
The seed is 14 The total is 100
The seed is 15 The total is 100
The seed is 16 The total is 100
The seed is 17 The total is 100
The seed is 18 The total is 100
The seed is 19 The total is 99
The seed is 20 The total is 100
The seed is 21 The total is 100
The seed is 22 The total is 100
The seed is 23 The total is 100
The seed is 24 The total is 100
The seed is 25 The total is 100
The seed is 26 The total is 100
The seed is 27 The total is 100
The seed is 28 The total is 100
The seed is 29 The total is 100
The seed is 30 The total is 100
The seed is 31 The total is 100
The seed is 32 The total is 100
The seed is 33 The total is 99
The seed is 34 The total is 100
The seed is 35 The total is 99
The seed is 36 The total is 100
The seed is 37 The total is 100
The seed is 38 The total is 100
The seed is 39 The total is 100
The seed is 40 The total is 97
The seed is 41 The total is 100
The seed is 42 The total is 100
The seed is 43 The total is 99
The seed is 44 The total is 100
The seed is 45 The total is 100
The seed is 46 The total is 99
The seed is 47 The total is 100
The seed is 48 The total is 100
The seed is 49 The total is 100
The seed is 50 The total is 100
The grand total for all trials is 4992

neural net approach known maximum
the size is 20 the multiplier is 2
the range is 20
The seed is 1 The total is 397
The seed is 2 The total is 399
The seed is 3 The total is 400
The seed is 4 The total is 399
The seed is 5 The total is 397
The seed is 6 The total is 399
The seed is 7 The total is 400
The seed is 8 The total is 400
The seed is 9 The total is 398
The seed is 10 The total is 397
The seed is 11 The total is 400
The seed is 12 The total is 400
The seed is 13 The total is 397
The seed is 14 The total is 400
The seed is 15 The total is 400
The seed is 16 The total is 398
The seed is 17 The total is 400
The seed is 18 The total is 400
The seed is 19 The total is 399
The seed is 20 The total is 397
The seed is 21 The total is 400
The seed is 22 The total is 400
The seed is 23 The total is 400
The seed is 24 The total is 398
The seed is 25 The total is 400
The seed is 26 The total is 398
The seed is 27 The total is 397
The seed is 28 The total is 399
The seed is 29 The total is 400
The seed is 30 The total is 400
The seed is 31 The total is 398
The seed is 32 The total is 398
The seed is 33 The total is 399
The seed is 34 The total is 399
The seed is 35 The total is 400
The seed is 36 The total is 400
The seed is 37 The total is 395
The seed is 38 The total is 397
The seed is 39 The total is 400
The seed is 40 The total is 395
The seed is 41 The total is 400
The seed is 42 The total is 400
The seed is 43 The total is 399
The seed is 44 The total is 399
The seed is 45 The total is 398
The seed is 46 The total is 398
The seed is 47 The total is 400
The seed is 48 The total is 400
The seed is 49 The total is 400
The seed is 50 The total is 399
The grand total for all trials is 19943

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
neural net approach known maximum
the size is 20 the multiplier is 3
the range is 20
The seed is 1 The total is 399
The seed is 2 The total is 400
The seed is 3 The total is 400
The seed is 4 The total is 399
The seed is 5 The total is 398
The seed is 6 The total is 399
The seed is 7 The total is 398
The seed is 8 The total is 400
The seed is 9 The total is 400
The seed is 10 The total is 398
The seed is 11 The total is 400
The seed is 12 The total is 400
The seed is 13 The total is 397
The seed is 14 The total is 400
The seed is 15 The total is 400
The seed is 16 The total is 400
The seed is 17 The total is 399
The seed is 18 The total is 400
The seed is 19 The total is 400
The seed is 20 The total is 398
The seed is 21 The total is 400
The seed is 22 The total is 400
The seed is 23 The total is 400
The seed is 24 The total is 400
The seed is 25 The total is 398
The seed is 26 The total is 398
The seed is 27 The total is 398
The seed is 28 The total is 399
The seed is 29 The total is 400
The seed is 30 The total is 397
The seed is 31 The total is 400
The seed is 32 The total is 398
The seed is 33 The total is 399
The seed is 34 The total is 400
The seed is 35 The total is 399
The seed is 36 The total is 400
The seed is 37 The total is 400
The seed is 38 The total is 400
The seed is 39 The total is 400
The seed is 40 The total is 395
The seed is 41 The total is 400
The seed is 42 The total is 400
The seed is 43 The total is 399
The seed is 44 The total is 400
The seed is 45 The total is 398
The seed is 46 The total is 398
The seed is 47 The total is 400
The seed is 48 The total is 398
The seed is 49 The total is 400
The seed is 50 The total is 399
The grand total for all trials is 19960

neural net approach known maximum
the size is 20 the multiplier is 5
the range is 20
The seed is 1 The total is 400
The seed is 2 The total is 399
The seed is 3 The total is 400
The seed is 4 The total is 399
The seed is 5 The total is 397
The seed is 6 The total is 400
The seed is 7 The total is 400
The seed is 8 The total is 400
The seed is 9 The total is 400
The seed is 10 The total is 400
The seed is 11 The total is 400
The seed is 12 The total is 400
The seed is 13 The total is 397
The seed is 14 The total is 400
The seed is 15 The total is 400
The seed is 16 The total is 397
The seed is 17 The total is 400
The seed is 18 The total is 400
The seed is 19 The total is 400
The seed is 20 The total is 400
The seed is 21 The total is 400
The seed is 22 The total is 400
The seed is 23 The total is 399
The seed is 24 The total is 400
The seed is 25 The total is 400
The seed is 26 The total is 400
The seed is 27 The total is 400
The seed is 28 The total is 400
The seed is 29 The total is 400
The seed is 30 The total is 400
The seed is 31 The total is 400
The seed is 32 The total is 400
The seed is 33 The total is 400
The seed is 34 The total is 400
The seed is 35 The total is 400
The seed is 36 The total is 400
The seed is 37 The total is 400
The seed is 38 The total is 400
The seed is 39 The total is 400
The seed is 40 The total is 398
The seed is 41 The total is 400
The seed is 42 The total is 400
The seed is 43 The total is 399
The seed is 44 The total is 400
The seed is 45 The total is 398
The seed is 46 The total is 398
The seed is 47 The total is 400
The seed is 48 The total is 398
The seed is 49 The total is 400
The seed is 50 The total is 399
The grand total for all trials is 19980

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
neural net approach known maximum
the size is 20 the multiplier is 10
the range is 20
The seed is 1 The total is 399
The seed is 2 The total is 399
The seed is 3 The total is 400
The seed is 4 The total is 399
The seed is 5 The total is 400
The seed is 6 The total is 400
The seed is 7 The total is 400
The seed is 8 The total is 400
The seed is 9 The total is 400
The seed is 10 The total is 400
The seed is 11 The total is 400
The seed is 12 The total is 400
The seed is 13 The total is 400
The seed is 14 The total is 400
The seed is 15 The total is 400
The seed is 16 The total is 400
The seed is 17 The total is 399
The seed is 18 The total is 400
The seed is 19 The total is 400
The seed is 20 The total is 400
The seed is 21 The total is 400
The seed is 22 The total is 400
The seed is 23 The total is 400
The seed is 24 The total is 400
The seed is 25 The total is 399
The seed is 26 The total is 399
The seed is 27 The total is 400
The seed is 28 The total is 400
The seed is 29 The total is 399
The seed is 30 The total is 399
The seed is 31 The total is 400
The seed is 32 The total is 400
The seed is 33 The total is 400
The seed is 34 The total is 400
The seed is 35 The total is 400
The seed is 36 The total is 398
The seed is 37 The total is 399
The seed is 38 The total is 400
The seed is 39 The total is 400
The seed is 40 The total is 400
The seed is 41 The total is 400
The seed is 42 The total is 400
The seed is 43 The total is 400
The seed is 44 The total is 400
The seed is 45 The total is 400
The seed is 46 The total is 397
The seed is 47 The total is 400
The seed is 48 The total is 400
The seed is 49 The total is 400
The seed is 50 The total is 400
The grand total for all trials is 19986

neural net approach known maximum
the size is 20 the multiplier is 10
the range is 20
The seed is 1  The total is 399
The seed is 2  The total is 399
The seed is 3  The total is 400
The seed is 4  The total is 399
The seed is 5  The total is 400
The seed is 6  The total is 400
The seed is 7  The total is 400
The seed is 8  The total is 400
The seed is 9  The total is 400
The seed is 10 The total is 400
The seed is 11 The total is 400
The seed is 12 The total is 400
The seed is 13 The total is 400
The seed is 14 The total is 400
The seed is 15 The total is 400
The seed is 16 The total is 400
The seed is 17 The total is 399
The seed is 18 The total is 400
The seed is 19 The total is 400
The seed is 20 The total is 400
The seed is 21 The total is 400
The seed is 22 The total is 400
The seed is 23 The total is 400
The seed is 24 The total is 400
The seed is 25 The total is 399
The seed is 26 The total is 399
The seed is 27 The total is 400
The seed is 28 The total is 400
The seed is 29 The total is 399
The seed is 30 The total is 399
The seed is 31 The total is 400
The seed is 32 The total is 400
The seed is 33 The total is 400
The seed is 34 The total is 400
The seed is 35 The total is 400
The seed is 36 The total is 398
The seed is 37 The total is 399
The seed is 38 The total is 400
The seed is 39 The total is 400
The seed is 40 The total is 400
The seed is 41 The total is 400
The seed is 42 The total is 400
The seed is 43 The total is 400
The seed is 44 The total is 400
The seed is 45 The total is 400
The seed is 46 The total is 397
The seed is 47 The total is 400
The seed is 48 The total is 400
The seed is 49 The total is 400
The seed is 50 The total is 400
The grand total for all trials is 19985
neural net approach unknown maximum
the size is 10 the multiplier is 3
the range is 10
The seed is 1 The total is 88
The seed is 2 The total is 92
The seed is 3 The total is 94
The seed is 4 The total is 85
The seed is 5 The total is 92
The seed is 6 The total is 94
The seed is 7 The total is 93
The seed is 8 The total is 80
The seed is 9 The total is 91
The seed is 10 The total is 90
The seed is 11 The total is 88
The seed is 12 The total is 92
The seed is 13 The total is 93
The seed is 14 The total is 94
The seed is 15 The total is 94
The seed is 16 The total is 91
The seed is 17 The total is 88
The seed is 18 The total is 91
The seed is 19 The total is 94
The seed is 20 The total is 94
The seed is 21 The total is 92
The seed is 22 The total is 82
The seed is 23 The total is 87
The seed is 24 The total is 83
The seed is 25 The total is 91
The seed is 26 The total is 94
The seed is 27 The total is 94
The seed is 28 The total is 88
The seed is 29 The total is 84
The seed is 30 The total is 85
The seed is 31 The total is 92
The seed is 32 The total is 94
The seed is 33 The total is 92
The seed is 34 The total is 89
The seed is 35 The total is 95
The seed is 36 The total is 89
The seed is 37 The total is 91
The seed is 38 The total is 97
The seed is 39 The total is 93
The seed is 40 The total is 93
The seed is 41 The total is 97
The seed is 42 The total is 89
The seed is 43 The total is 86
The seed is 44 The total is 95
The seed is 45 The total is 90
The seed is 46 The total is 95
The seed is 47 The total is 88
The seed is 48 The total is 94
The seed is 49 The total is 89
The seed is 50 The total is 93
The grand total for all trials is 4537

neural net approach unknown maximum
the size is 10 the multiplier is 5
the range is 10
The seed is 1 The total is 88
The seed is 2 The total is 92
The seed is 3 The total is 94
The seed is 4 The total is 85
The seed is 5 The total is 92
The seed is 6 The total is 94
The seed is 7 The total is 93
The seed is 8 The total is 80
The seed is 9 The total is 91
The seed is 10 The total is 90
The seed is 11 The total is 88
The seed is 12 The total is 91
The seed is 13 The total is 93
The seed is 14 The total is 94
The seed is 15 The total is 94
The seed is 16 The total is 91
The seed is 17 The total is 88
The seed is 18 The total is 91
The seed is 19 The total is 96
The seed is 20 The total is 92
The seed is 21 The total is 92
The seed is 22 The total is 82
The seed is 23 The total is 87
The seed is 24 The total is 83
The seed is 25 The total is 91
The seed is 26 The total is 94
The seed is 27 The total is 94
The seed is 28 The total is 88
The seed is 29 The total is 84
The seed is 30 The total is 85
The seed is 31 The total is 92
The seed is 32 The total is 94
The seed is 33 The total is 92
The seed is 34 The total is 89
The seed is 35 The total is 95
The seed is 36 The total is 89
The seed is 37 The total is 91
The seed is 38 The total is 97
The seed is 39 The total is 93
The seed is 40 The total is 92
The seed is 41 The total is 97
The seed is 42 The total is 89
The seed is 43 The total is 86
The seed is 44 The total is 95
The seed is 45 The total is 90
The seed is 46 The total is 95
The seed is 47 The total is 88
The seed is 48 The total is 94
The seed is 49 The total is 89
The seed is 50 The total is 94
The grand total for all trials is 4535
neural net approach unknown maximum
the size is 10 the multiplier is 10
the range is 10
The seed is 1 The total is 86
The seed is 2 The total is 92
The seed is 3 The total is 94
The seed is 4 The total is 85
The seed is 5 The total is 92
The seed is 6 The total is 94
The seed is 7 The total is 93
The seed is 8 The total is 80
The seed is 9 The total is 91
The seed is 10 The total is 90
The seed is 11 The total is 88
The seed is 12 The total is 91
The seed is 13 The total is 93
The seed is 14 The total is 94
The seed is 15 The total is 94
The seed is 16 The total is 91
The seed is 17 The total is 88
The seed is 18 The total is 91
The seed is 19 The total is 96
The seed is 20 The total is 94
The seed is 21 The total is 92
The seed is 22 The total is 92
The seed is 23 The total is 82
The seed is 24 The total is 87
The seed is 25 The total is 85
The seed is 26 The total is 91
The seed is 27 The total is 94
The seed is 28 The total is 88
The seed is 29 The total is 84
The seed is 30 The total is 85
The seed is 31 The total is 92
The seed is 32 The total is 94
The seed is 33 The total is 92
The seed is 34 The total is 89
The seed is 35 The total is 95
The seed is 36 The total is 88
The seed is 37 The total is 91
The seed is 38 The total is 97
The seed is 39 The total is 91
The seed is 40 The total is 92
The seed is 41 The total is 97
The seed is 42 The total is 87
The seed is 43 The total is 85
The seed is 44 The total is 95
The seed is 45 The total is 90
The seed is 46 The total is 95
The seed is 47 The total is 88
The seed is 48 The total is 94
The seed is 49 The total is 89
The seed is 50 The total is 94
The grand total for all trials is 4530

neural net approach unknown maximum
the size is 10 the multiplier is 20
the range is 10
The seed is 1 The total is 86
The seed is 2 The total is 92
The seed is 3 The total is 94
The seed is 4 The total is 85
The seed is 5 The total is 92
The seed is 6 The total is 94
The seed is 7 The total is 93
The seed is 8 The total is 80
The seed is 9 The total is 91
The seed is 10 The total is 90
The seed is 11 The total is 88
The seed is 12 The total is 91
The seed is 13 The total is 93
The seed is 14 The total is 94
The seed is 15 The total is 94
The seed is 16 The total is 91
The seed is 17 The total is 88
The seed is 18 The total is 91
The seed is 19 The total is 96
The seed is 20 The total is 94
The seed is 21 The total is 92
The seed is 22 The total is 92
The seed is 23 The total is 82
The seed is 24 The total is 87
The seed is 25 The total is 85
The seed is 26 The total is 91
The seed is 27 The total is 94
The seed is 28 The total is 88
The seed is 29 The total is 84
The seed is 30 The total is 85
The seed is 31 The total is 92
The seed is 32 The total is 94
The seed is 33 The total is 92
The seed is 34 The total is 89
The seed is 35 The total is 95
The seed is 36 The total is 88
The seed is 37 The total is 91
The seed is 38 The total is 97
The seed is 39 The total is 91
The seed is 40 The total is 92
The seed is 41 The total is 97
The seed is 42 The total is 87
The seed is 43 The total is 85
The seed is 44 The total is 95
The seed is 45 The total is 90
The seed is 46 The total is 95
The seed is 47 The total is 88
The seed is 48 The total is 94
The seed is 49 The total is 89
The seed is 50 The total is 94
The grand total for all trials is 4530

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
The grand total for all trials is 18852.
<table>
<thead>
<tr>
<th>Neural Net Approach Unknown Maximum</th>
<th>Neural Net Approach Unknown Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Size is 20, The Multiplier is 5</td>
<td>The Size is 20, The Multiplier is 10</td>
</tr>
<tr>
<td>The range is 20</td>
<td>The range is 20</td>
</tr>
<tr>
<td>The seed is 1, The total is 381</td>
<td>The seed is 1, The total is 381</td>
</tr>
<tr>
<td>The seed is 2, The total is 380</td>
<td>The seed is 2, The total is 380</td>
</tr>
<tr>
<td>The seed is 3, The total is 370</td>
<td>The seed is 3, The total is 370</td>
</tr>
<tr>
<td>The seed is 4, The total is 383</td>
<td>The seed is 4, The total is 383</td>
</tr>
<tr>
<td>The seed is 5, The total is 383</td>
<td>The seed is 5, The total is 383</td>
</tr>
<tr>
<td>The seed is 6, The total is 381</td>
<td>The seed is 6, The total is 381</td>
</tr>
<tr>
<td>The seed is 7, The total is 376</td>
<td>The seed is 7, The total is 376</td>
</tr>
<tr>
<td>The seed is 8, The total is 365</td>
<td>The seed is 8, The total is 365</td>
</tr>
<tr>
<td>The seed is 9, The total is 378</td>
<td>The seed is 9, The total is 378</td>
</tr>
<tr>
<td>The seed is 10, The total is 379</td>
<td>The seed is 10, The total is 379</td>
</tr>
<tr>
<td>The seed is 11, The total is 373</td>
<td>The seed is 11, The total is 373</td>
</tr>
<tr>
<td>The seed is 12, The total is 374</td>
<td>The seed is 12, The total is 374</td>
</tr>
<tr>
<td>The seed is 13, The total is 367</td>
<td>The seed is 13, The total is 364</td>
</tr>
<tr>
<td>The seed is 14, The total is 372</td>
<td>The seed is 14, The total is 372</td>
</tr>
<tr>
<td>The seed is 15, The total is 370</td>
<td>The seed is 15, The total is 372</td>
</tr>
<tr>
<td>The seed is 16, The total is 377</td>
<td>The seed is 16, The total is 376</td>
</tr>
<tr>
<td>The seed is 17, The total is 376</td>
<td>The seed is 17, The total is 376</td>
</tr>
<tr>
<td>The seed is 18, The total is 362</td>
<td>The seed is 18, The total is 362</td>
</tr>
<tr>
<td>The seed is 19, The total is 376</td>
<td>The seed is 19, The total is 376</td>
</tr>
<tr>
<td>The seed is 20, The total is 379</td>
<td>The seed is 20, The total is 379</td>
</tr>
<tr>
<td>The seed is 21, The total is 382</td>
<td>The seed is 21, The total is 377</td>
</tr>
<tr>
<td>The seed is 22, The total is 375</td>
<td>The seed is 22, The total is 375</td>
</tr>
<tr>
<td>The seed is 23, The total is 378</td>
<td>The seed is 23, The total is 376</td>
</tr>
<tr>
<td>The seed is 24, The total is 379</td>
<td>The seed is 24, The total is 375</td>
</tr>
<tr>
<td>The seed is 25, The total is 380</td>
<td>The seed is 25, The total is 380</td>
</tr>
<tr>
<td>The seed is 26, The total is 378</td>
<td>The seed is 26, The total is 378</td>
</tr>
<tr>
<td>The seed is 27, The total is 378</td>
<td>The seed is 27, The total is 377</td>
</tr>
<tr>
<td>The seed is 28, The total is 375</td>
<td>The seed is 28, The total is 375</td>
</tr>
<tr>
<td>The seed is 29, The total is 380</td>
<td>The seed is 29, The total is 380</td>
</tr>
<tr>
<td>The seed is 30, The total is 370</td>
<td>The seed is 30, The total is 375</td>
</tr>
<tr>
<td>The seed is 31, The total is 367</td>
<td>The seed is 31, The total is 363</td>
</tr>
<tr>
<td>The seed is 32, The total is 369</td>
<td>The seed is 32, The total is 369</td>
</tr>
<tr>
<td>The seed is 33, The total is 373</td>
<td>The seed is 33, The total is 389</td>
</tr>
<tr>
<td>The seed is 34, The total is 374</td>
<td>The seed is 34, The total is 373</td>
</tr>
<tr>
<td>The seed is 35, The total is 387</td>
<td>The seed is 35, The total is 372</td>
</tr>
<tr>
<td>The seed is 36, The total is 379</td>
<td>The seed is 36, The total is 386</td>
</tr>
<tr>
<td>The seed is 37, The total is 378</td>
<td>The seed is 37, The total is 378</td>
</tr>
<tr>
<td>The seed is 38, The total is 396</td>
<td>The seed is 38, The total is 396</td>
</tr>
<tr>
<td>The seed is 39, The total is 392</td>
<td>The seed is 39, The total is 391</td>
</tr>
<tr>
<td>The seed is 40, The total is 369</td>
<td>The seed is 40, The total is 378</td>
</tr>
<tr>
<td>The seed is 41, The total is 385</td>
<td>The seed is 41, The total is 381</td>
</tr>
<tr>
<td>The seed is 42, The total is 380</td>
<td>The seed is 42, The total is 380</td>
</tr>
<tr>
<td>The seed is 43, The total is 374</td>
<td>The seed is 43, The total is 374</td>
</tr>
<tr>
<td>The seed is 44, The total is 374</td>
<td>The seed is 44, The total is 366</td>
</tr>
<tr>
<td>The seed is 45, The total is 384</td>
<td>The seed is 45, The total is 384</td>
</tr>
<tr>
<td>The seed is 46, The total is 372</td>
<td>The seed is 46, The total is 372</td>
</tr>
<tr>
<td>The seed is 47, The total is 364</td>
<td>The seed is 47, The total is 354</td>
</tr>
<tr>
<td>The seed is 48, The total is 362</td>
<td>The seed is 48, The total is 382</td>
</tr>
<tr>
<td>The seed is 49, The total is 360</td>
<td>The seed is 49, The total is 363</td>
</tr>
<tr>
<td>The seed is 50, The total is 380</td>
<td>The seed is 50, The total is 375</td>
</tr>
</tbody>
</table>

The grand total for all trials is 18832

The grand total for all trials is 18807

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
neural net approach unknown maximum
the size is 20 the multiplier is 20
the range is 20
The seed is 1 The total is 381
The seed is 2 The total is 380
The seed is 3 The total is 370
The seed is 4 The total is 383
The seed is 5 The total is 383
The seed is 6 The total is 381
The seed is 7 The total is 373
The seed is 8 The total is 365
The seed is 9 The total is 376
The seed is 10 The total is 379
The seed is 11 The total is 371
The seed is 12 The total is 374
The seed is 13 The total is 364
The seed is 14 The total is 372
The seed is 15 The total is 372
The seed is 16 The total is 376
The seed is 17 The total is 380
The seed is 18 The total is 362
The seed is 19 The total is 373
The seed is 20 The total is 378
The seed is 21 The total is 374
The seed is 22 The total is 378
The seed is 23 The total is 376
The seed is 24 The total is 379
The seed is 25 The total is 380
The seed is 26 The total is 380
The seed is 27 The total is 377
The seed is 28 The total is 375
The seed is 29 The total is 380
The seed is 30 The total is 375
The seed is 31 The total is 363
The seed is 32 The total is 369
The seed is 33 The total is 373
The seed is 34 The total is 374
The seed is 35 The total is 386
The seed is 36 The total is 378
The seed is 37 The total is 378
The seed is 38 The total is 396
The seed is 39 The total is 391
The seed is 40 The total is 378
The seed is 41 The total is 381
The seed is 42 The total is 360
The seed is 43 The total is 374
The seed is 44 The total is 366
The seed is 45 The total is 384
The seed is 46 The total is 372
The seed is 47 The total is 354
The seed is 48 The total is 352
The seed is 49 The total is 363
The seed is 50 The total is 375
The grand total for all trials is 18806

neural net approach unknown maximum
the size is 13 the multiplier is 2
the range is 13
The seed is 1 The total is 154
The seed is 2 The total is 153
The seed is 3 The total is 153
The seed is 4 The total is 150
The seed is 5 The total is 163
The seed is 6 The total is 156
The seed is 7 The total is 154
The seed is 8 The total is 154
The seed is 9 The total is 157
The seed is 10 The total is 157
The seed is 11 The total is 155
The seed is 12 The total is 152
The seed is 13 The total is 142
The seed is 14 The total is 165
The seed is 15 The total is 157
The seed is 16 The total is 156
The seed is 17 The total is 156
The seed is 18 The total is 156
The seed is 19 The total is 150
The seed is 20 The total is 155
The seed is 21 The total is 153
The seed is 22 The total is 154
The seed is 23 The total is 156
The seed is 24 The total is 154
The seed is 25 The total is 156
The seed is 26 The total is 146
The seed is 27 The total is 157
The seed is 28 The total is 160
The seed is 29 The total is 156
The seed is 30 The total is 159
The seed is 31 The total is 159
The seed is 32 The total is 161
The seed is 33 The total is 143
The seed is 34 The total is 153
The seed is 35 The total is 152
The seed is 36 The total is 158
The seed is 37 The total is 152
The seed is 38 The total is 155
The seed is 39 The total is 158
The seed is 40 The total is 153
The seed is 41 The total is 151
The seed is 42 The total is 149
The seed is 43 The total is 154
The seed is 44 The total is 157
The seed is 45 The total is 154
The seed is 46 The total is 155
The seed is 47 The total is 159
The seed is 48 The total is 156
The seed is 49 The total is 151
The seed is 50 The total is 154
The grand total for all trials is 7733

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
neural net approach unknown maximum
the size is 13 the multiplier is 2
the range is 13

The seed is 51 The total is 155
The seed is 52 The total is 155
The seed is 53 The total is 146
The seed is 54 The total is 154
The seed is 55 The total is 151
The seed is 56 The total is 160
The seed is 57 The total is 153
The seed is 58 The total is 158
The seed is 59 The total is 154
The seed is 60 The total is 146
The seed is 61 The total is 157
The seed is 62 The total is 157
The seed is 63 The total is 159
The seed is 64 The total is 157
The seed is 65 The total is 150
The seed is 66 The total is 161
The seed is 67 The total is 153
The seed is 68 The total is 157
The seed is 69 The total is 158
The seed is 70 The total is 152
The seed is 71 The total is 160
The seed is 72 The total is 151
The seed is 73 The total is 164
The seed is 74 The total is 157
The seed is 75 The total is 153
The seed is 76 The total is 158
The seed is 77 The total is 163
The seed is 78 The total is 152
The seed is 79 The total is 152
The seed is 80 The total is 161
The seed is 81 The total is 158
The seed is 82 The total is 155
The seed is 83 The total is 157
The seed is 84 The total is 151
The seed is 85 The total is 156
The seed is 86 The total is 154
The seed is 87 The total is 156
The seed is 88 The total is 156
The seed is 89 The total is 149
The seed is 90 The total is 156
The seed is 91 The total is 150
The seed is 92 The total is 160
The seed is 93 The total is 160
The seed is 94 The total is 159
The seed is 95 The total is 156
The seed is 96 The total is 154
The seed is 97 The total is 151
The seed is 98 The total is 157
The seed is 99 The total is 155
The seed is 100 The total is 153
The grand total for all trials is 7767

The seed is 101 The total is 152
The seed is 102 The total is 159
The seed is 103 The total is 155
The seed is 104 The total is 150
The seed is 105 The total is 159
The seed is 106 The total is 159
The seed is 107 The total is 156
The seed is 108 The total is 161
The seed is 109 The total is 145
The seed is 110 The total is 156
The seed is 111 The total is 156
The seed is 112 The total is 156
The seed is 113 The total is 160
The seed is 114 The total is 154
The seed is 115 The total is 140
The seed is 116 The total is 155
The seed is 117 The total is 154
The seed is 118 The total is 158
The seed is 119 The total is 164
The seed is 120 The total is 155
The seed is 121 The total is 156
The seed is 122 The total is 157
The seed is 123 The total is 158
The seed is 124 The total is 162
The seed is 125 The total is 158
The seed is 126 The total is 162
The seed is 127 The total is 152
The seed is 128 The total is 156
The seed is 129 The total is 161
The seed is 130 The total is 155
The seed is 131 The total is 157
The seed is 132 The total is 149
The seed is 133 The total is 153
The seed is 134 The total is 150
The seed is 135 The total is 150
The seed is 136 The total is 153
The seed is 137 The total is 163
The seed is 138 The total is 158
The seed is 139 The total is 161
The seed is 140 The total is 152
The seed is 141 The total is 153
The seed is 142 The total is 155
The seed is 143 The total is 152
The seed is 144 The total is 161
The seed is 145 The total is 158
The seed is 146 The total is 157
The seed is 147 The total is 158
The seed is 148 The total is 158
The seed is 149 The total is 154
The seed is 150 The total is 151
The grand total for all trials is 7784
neural net approach unknown maximum
the size is 13 the multiplier is 2
the range is 26
The seed is 151 The total is 297
The seed is 152 The total is 311
The seed is 153 The total is 318
The seed is 154 The total is 308
The seed is 155 The total is 314
The seed is 156 The total is 301
The seed is 157 The total is 293
The seed is 158 The total is 307
The seed is 159 The total is 296
The seed is 160 The total is 302
The seed is 161 The total is 307
The seed is 162 The total is 295
The seed is 163 The total is 299
The seed is 164 The total is 304
The seed is 165 The total is 310
The seed is 166 The total is 305
The seed is 167 The total is 304
The seed is 168 The total is 304
The seed is 169 The total is 286
The seed is 170 The total is 308
The seed is 171 The total is 293
The seed is 172 The total is 303
The seed is 173 The total is 318
The seed is 174 The total is 296
The seed is 175 The total is 311
The seed is 176 The total is 305
The seed is 177 The total is 306
The seed is 178 The total is 302
The seed is 179 The total is 310
The seed is 180 The total is 299
The seed is 181 The total is 307
The seed is 182 The total is 303
The seed is 183 The total is 298
The seed is 184 The total is 301
The seed is 185 The total is 308
The seed is 186 The total is 314
The seed is 187 The total is 312
The seed is 188 The total is 299
The seed is 189 The total is 287
The seed is 190 The total is 301
The seed is 191 The total is 314
The seed is 192 The total is 307
The seed is 193 The total is 297
The seed is 194 The total is 308
The seed is 195 The total is 307
The seed is 196 The total is 309
The seed is 197 The total is 301
The seed is 198 The total is 299
The seed is 199 The total is 313
The seed is 200 The total is 293
The grand total for all trials is 15192

neural net approach unknown maximum
the size is 13 the multiplier is 2
the range is 26
The seed is 201 The total is 309
The seed is 202 The total is 310
The seed is 203 The total is 299
The seed is 204 The total is 279
The seed is 205 The total is 313
The seed is 206 The total is 310
The seed is 207 The total is 294
The seed is 208 The total is 306
The seed is 209 The total is 313
The seed is 210 The total is 295
The seed is 211 The total is 312
The seed is 212 The total is 311
The seed is 213 The total is 301
The seed is 214 The total is 303
The seed is 215 The total is 307
The seed is 216 The total is 306
The seed is 217 The total is 306
The seed is 218 The total is 307
The seed is 219 The total is 314
The seed is 220 The total is 310
The seed is 221 The total is 310
The seed is 222 The total is 320
The seed is 223 The total is 301
The seed is 224 The total is 318
The seed is 225 The total is 310
The seed is 226 The total is 308
The seed is 227 The total is 303
The seed is 228 The total is 325
The seed is 229 The total is 315
The seed is 230 The total is 313
The seed is 231 The total is 291
The seed is 232 The total is 316
The seed is 233 The total is 304
The seed is 234 The total is 303
The seed is 235 The total is 288
The seed is 236 The total is 306
The seed is 237 The total is 307
The seed is 238 The total is 314
The seed is 239 The total is 310
The seed is 240 The total is 319
The seed is 241 The total is 297
The seed is 242 The total is 317
The seed is 243 The total is 305
The seed is 244 The total is 292
The seed is 245 The total is 297
The seed is 246 The total is 309
The seed is 247 The total is 315
The seed is 248 The total is 298
The seed is 249 The total is 296
The seed is 250 The total is 312
The grand total for all trials is 15324

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
neural net approach unknown maximum
the size is 13 the multiplier is 2
the range is 26
The seed is 251 The total is 314
The seed is 252 The total is 299
The seed is 253 The total is 310
The seed is 254 The total is 320
The seed is 255 The total is 299
The seed is 256 The total is 307
The seed is 257 The total is 314
The seed is 258 The total is 307
The seed is 259 The total is 307
The seed is 260 The total is 286
The seed is 261 The total is 307
The seed is 262 The total is 298
The seed is 263 The total is 318
The seed is 264 The total is 310
The seed is 265 The total is 305
The seed is 266 The total is 306
The seed is 267 The total is 311
The seed is 268 The total is 311
The seed is 269 The total is 307
The seed is 270 The total is 306
The seed is 271 The total is 301
The seed is 272 The total is 297
The seed is 273 The total is 303
The seed is 274 The total is 307
The seed is 275 The total is 304
The seed is 276 The total is 298
The seed is 277 The total is 314
The seed is 278 The total is 301
The seed is 279 The total is 288
The seed is 280 The total is 276
The seed is 281 The total is 315
The seed is 282 The total is 310
The seed is 283 The total is 311
The seed is 284 The total is 308
The seed is 285 The total is 317
The seed is 286 The total is 321
The seed is 287 The total is 305
The seed is 288 The total is 320
The seed is 289 The total is 301
The seed is 290 The total is 298
The seed is 291 The total is 310
The seed is 292 The total is 315
The seed is 293 The total is 308
The seed is 294 The total is 305
The seed is 295 The total is 307
The seed is 296 The total is 319
The seed is 297 The total is 294
The seed is 298 The total is 311
The seed is 299 The total is 315
The seed is 300 The total is 282
The grand total for all trials is 15303

neural net approach unknown maximum
the size is 15 the multiplier is 2
the range is 7
The seed is 301 The total is 83
The seed is 302 The total is 82
The seed is 303 The total is 83
The seed is 304 The total is 86
The seed is 305 The total is 82
The seed is 306 The total is 84
The seed is 307 The total is 83
The seed is 308 The total is 81
The seed is 309 The total is 86
The seed is 310 The total is 80
The seed is 311 The total is 82
The seed is 312 The total is 80
The seed is 313 The total is 83
The seed is 314 The total is 83
The seed is 315 The total is 85
The seed is 316 The total is 81
The seed is 317 The total is 77
The seed is 318 The total is 85
The seed is 319 The total is 84
The seed is 320 The total is 80
The seed is 321 The total is 81
The seed is 322 The total is 82
The seed is 323 The total is 80
The seed is 324 The total is 82
The seed is 325 The total is 86
The seed is 326 The total is 80
The seed is 327 The total is 84
The seed is 328 The total is 87
The seed is 329 The total is 82
The seed is 330 The total is 85
The seed is 331 The total is 85
The seed is 332 The total is 81
The seed is 333 The total is 82
The seed is 334 The total is 86
The seed is 335 The total is 80
The seed is 336 The total is 81
The seed is 337 The total is 79
The seed is 338 The total is 81
The seed is 339 The total is 82
The seed is 340 The total is 82
The seed is 341 The total is 86
The seed is 342 The total is 82
The seed is 343 The total is 79
The seed is 344 The total is 82
The seed is 345 The total is 85
The seed is 346 The total is 84
The seed is 347 The total is 80
The seed is 348 The total is 80
The seed is 349 The total is 81
The seed is 350 The total is 85
The grand total for all trials is 4124
neural net approach unknown maximum
the size is 13 the multiplier is 2
the range is 7
The seed is 351 The total is 85
The seed is 352 The total is 85
The seed is 353 The total is 81
The seed is 354 The total is 82
The seed is 355 The total is 80
The seed is 356 The total is 79
The seed is 357 The total is 84
The seed is 358 The total is 84
The seed is 359 The total is 80
The seed is 360 The total is 84
The seed is 361 The total is 86
The seed is 362 The total is 81
The seed is 363 The total is 83
The seed is 364 The total is 84
The seed is 365 The total is 81
The seed is 366 The total is 80
The seed is 367 The total is 84
The seed is 368 The total is 80
The seed is 369 The total is 83
The seed is 370 The total is 81
The seed is 371 The total is 83
The seed is 372 The total is 87
The seed is 373 The total is 83
The seed is 374 The total is 87
The seed is 375 The total is 79
The seed is 376 The total is 75
The seed is 377 The total is 78
The seed is 378 The total is 83
The seed is 379 The total is 83
The seed is 380 The total is 86
The seed is 381 The total is 83
The seed is 382 The total is 82
The seed is 383 The total is 87
The seed is 384 The total is 81
The seed is 385 The total is 85
The seed is 386 The total is 81
The seed is 387 The total is 83
The seed is 388 The total is 83
The seed is 389 The total is 80
The seed is 390 The total is 88
The seed is 391 The total is 83
The seed is 392 The total is 84
The seed is 393 The total is 79
The seed is 394 The total is 86
The seed is 395 The total is 83
The seed is 396 The total is 85
The seed is 397 The total is 85
The seed is 398 The total is 79
The seed is 399 The total is 80
The seed is 400 The total is 82
The grand total for all trials is 4130

neural net approach unknown maximum
the size is 20 the multiplier is 2
the range is 20
The seed is 401 The total is 377
The seed is 402 The total is 382
The seed is 403 The total is 370
The seed is 404 The total is 370
The seed is 405 The total is 383
The seed is 406 The total is 379
The seed is 407 The total is 374
The seed is 408 The total is 375
The seed is 409 The total is 368
The seed is 410 The total is 378
The seed is 411 The total is 378
The seed is 412 The total is 392
The seed is 413 The total is 377
The seed is 414 The total is 374
The seed is 415 The total is 374
The seed is 416 The total is 372
The seed is 417 The total is 378
The seed is 418 The total is 379
The seed is 419 The total is 384
The seed is 420 The total is 369
The seed is 421 The total is 378
The seed is 422 The total is 375
The seed is 423 The total is 380
The seed is 424 The total is 378
The seed is 425 The total is 374
The seed is 426 The total is 386
The seed is 427 The total is 376
The seed is 428 The total is 383
The seed is 429 The total is 380
The seed is 430 The total is 383
The seed is 431 The total is 361
The seed is 432 The total is 379
The seed is 433 The total is 374
The seed is 434 The total is 378
The seed is 435 The total is 380
The seed is 436 The total is 388
The seed is 437 The total is 374
The seed is 438 The total is 379
The seed is 439 The total is 379
The seed is 440 The total is 374
The seed is 441 The total is 384
The seed is 442 The total is 375
The seed is 443 The total is 372
The seed is 444 The total is 386
The seed is 445 The total is 380
The seed is 446 The total is 370
The seed is 447 The total is 384
The seed is 448 The total is 376
The seed is 449 The total is 376
The seed is 450 The total is 381
The grand total for all trials is 18876
neural net approach unknown maximum
the size is 20 the multiplier is 2
the range is 20

The seed is 451 The total is 376
The seed is 452 The total is 378
The seed is 453 The total is 381
The seed is 454 The total is 387
The seed is 455 The total is 375
The seed is 456 The total is 378
The seed is 457 The total is 384
The seed is 458 The total is 375
The seed is 459 The total is 379
The seed is 460 The total is 376
The seed is 461 The total is 388
The seed is 462 The total is 381
The seed is 463 The total is 392
The seed is 464 The total is 371
The seed is 465 The total is 377
The seed is 466 The total is 368
The seed is 467 The total is 371
The seed is 468 The total is 380
The seed is 469 The total is 378
The seed is 470 The total is 376
The seed is 471 The total is 385
The seed is 472 The total is 375
The seed is 473 The total is 385
The seed is 474 The total is 370
The seed is 475 The total is 373
The seed is 476 The total is 365
The seed is 477 The total is 365
The seed is 478 The total is 383
The seed is 479 The total is 376
The seed is 480 The total is 380
The seed is 481 The total is 375
The seed is 482 The total is 367
The seed is 483 The total is 374
The seed is 484 The total is 386
The seed is 485 The total is 381
The seed is 486 The total is 384
The seed is 487 The total is 380
The seed is 488 The total is 379
The seed is 489 The total is 368
The seed is 490 The total is 375
The seed is 491 The total is 380
The seed is 492 The total is 375
The seed is 493 The total is 379
The seed is 494 The total is 380
The seed is 495 The total is 368
The seed is 496 The total is 383
The seed is 497 The total is 383
The seed is 498 The total is 375
The seed is 499 The total is 384
The seed is 500 The total is 379

The grand total for all trials is 18891

neural net approach unknown maximum
the size is 8 the multiplier is 2
the range is 8

The seed is 501 The total is 60
The seed is 502 The total is 52
The seed is 503 The total is 55
The seed is 504 The total is 62
The seed is 505 The total is 61
The seed is 506 The total is 56
The seed is 507 The total is 53
The seed is 508 The total is 59
The seed is 509 The total is 60
The seed is 510 The total is 62
The seed is 511 The total is 61
The seed is 512 The total is 58
The seed is 513 The total is 56
The seed is 514 The total is 53
The seed is 515 The total is 60
The seed is 516 The total is 55
The seed is 517 The total is 60
The seed is 518 The total is 59
The seed is 519 The total is 55
The seed is 520 The total is 57
The seed is 521 The total is 54
The seed is 522 The total is 53
The seed is 523 The total is 57
The seed is 524 The total is 57
The seed is 525 The total is 56
The seed is 526 The total is 55
The seed is 527 The total is 53
The seed is 528 The total is 61
The seed is 529 The total is 55
The seed is 530 The total is 60
The seed is 531 The total is 61
The seed is 532 The total is 61
The seed is 533 The total is 56
The seed is 534 The total is 58
The seed is 535 The total is 54
The seed is 536 The total is 57
The seed is 537 The total is 54
The seed is 538 The total is 55
The seed is 539 The total is 55
The seed is 540 The total is 54
The seed is 541 The total is 59
The seed is 542 The total is 62
The seed is 543 The total is 53
The seed is 544 The total is 58
The seed is 545 The total is 56
The seed is 546 The total is 60
The seed is 547 The total is 55
The seed is 548 The total is 58
The seed is 549 The total is 62
The seed is 550 The total is 60

The grand total for all trials is 2863
neural net approach unknown maximum
the size is 8 the multiplier is 2
the range is 8
The seed is 551 The total is 54
The seed is 552 The total is 59
The seed is 553 The total is 56
The seed is 554 The total is 58
The seed is 555 The total is 59
The seed is 556 The total is 54
The seed is 557 The total is 54
The seed is 558 The total is 58
The seed is 559 The total is 57
The seed is 560 The total is 59
The seed is 561 The total is 50
The seed is 562 The total is 57
The seed is 563 The total is 60
The seed is 564 The total is 59
The seed is 565 The total is 57
The seed is 566 The total is 55
The seed is 567 The total is 56
The seed is 568 The total is 59
The seed is 569 The total is 55
The seed is 570 The total is 61
The seed is 571 The total is 54
The seed is 572 The total is 54
The seed is 573 The total is 59
The seed is 574 The total is 55
The seed is 575 The total is 51
The seed is 576 The total is 61
The seed is 577 The total is 52
The seed is 578 The total is 54
The seed is 579 The total is 54
The seed is 580 The total is 60
The seed is 581 The total is 60
The seed is 582 The total is 55
The seed is 583 The total is 55
The seed is 584 The total is 60
The seed is 585 The total is 61
The seed is 586 The total is 58
The seed is 587 The total is 60
The seed is 588 The total is 55
The seed is 589 The total is 61
The seed is 590 The total is 63
The seed is 591 The total is 58
The seed is 592 The total is 53
The seed is 593 The total is 56
The seed is 594 The total is 61
The seed is 595 The total is 55
The seed is 596 The total is 54
The seed is 597 The total is 58
The seed is 598 The total is 58
The seed is 599 The total is 57
The seed is 600 The total is 53
The grand total for all trials is 2842

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
BIBLIOGRAPHY


