Mathematics Education and Graph Theory

Proceedings of International Seminar on Mathematics Education and Graph Theory
June 9, 2014

Editors:
Mustangin
Abdul Halim Fathani

Mathematics Education and Graph Theory

PROCEEDINGS OF INTERNATIONAL SEMINAR
ON MATHEMATICS EDUCATION AND GRAPH THEORY

Unit of Publication
Department of Mathematics Education
Faculty of Teacher Training and Education
Islamic University of Malang (UNISMA)
2014

MATHEMATICS EDUCATION AND GRAPH THEORY
Proceedings of International Seminar on Mathematics Education and Graph Theory

These proceedings contain the full texts of paper and talks presented in the International Seminar on Mathematics Education and Graph Theory on June 9, 2014

Reviewers
Surahmat Supangken (UNISMA, Indonesia)
Abdur Rahman As'ari (UM, Indonesia)
Kiki Ariyanti Sugeng (UI, Indonesia)
Sunismi (UNISMA, Indonesia)
Akhsanul In'am (UMM, Indonesia)

Editors
Mustangin
Abdul Halim Fathani

Layouter
Teguh Permadi

First published, 2014
ISBN 978-602-71141-0-4

Published by

Unit of Publication
Department of Mathematics Education
Faculty of Teacher Training and Education
Islamic University of Malang (UNISMA)
Malang, East Java, Indonesia
Phone +62341- 551932, 551822.
Fax +62341-552249
http://www.unisma.ac.id

PREFACE

These proceedings contain the full text of papers and talks presented in the International Seminar on Mathematics Education and Graph Theory. This seminar was held in conjunction with the International Workshop on Graph Masters. The workshop was held on June 7–8, 2014, while the seminar was on June 9, 2014. These events were organized by Islamic University of Malang (Unisma) in cooperation with Indonesian Combinatorial Society (InaCombS).

The workshop and the seminar would not have been possible without the time and energy put forth by the invited speakers. The invited speakers of the workshop were: **Mirka Miller**, University of Newcastle, Australia; **Joseph Miret**, Universitat de Lleida, Spain; **Christian Mauduit**, Institut de Mathematiques de Luminy, France; **Edy T. Baskoro**, Bandung Institute of Technology, Indonesia; **Surahmat Supangken**, Islamic University of Malang, Indonesia; **Tri Atmojo**, State University of Semarang, Indonesia; and **Purwanto**, State University of Malang, Indonesia.

The invited speakers of the seminar were: **Juddy Anne Osborn,** University of Newcastle, Australia and **Abdur Rahman As'ari**, State University of Malang, Indonesia. The seminar was held on the area of mathematics education and graph theory. The main themes of the mathematics education seminar include topics within the following areas (but not limited to): philosophy of mathematics education, curriculum development, learning methods and strategies, learning media, development of teaching material, and assessment and evaluation of learning. The main themes covered in graph theory seminar include topics within the following areas (but not limited to): degree (diameter) problems, ramsey numbers, cycles in graphs, graph labeling, dimensions of graphs, graph coloring, algorithmic graph theory, and applications of graph theory in various fields.

We would like to thank you to the invited speakers and all presenters who have submitted papers, for their valuable and inspiring presentation. A special appreciation goes to: **Surahmat Supangken**, Rector of Unisma and **Kiki Ariyanti Sugeng,** the President of InaCombS, who have made a lot of efforts to prepare this seminar.

We also do not forget to express our gratitude to Islamic University of Malang (Unisma) for providing financial support, and to the Indonesian Combinatorial Society (InaCombS) for the support. We hope that you had a great time and valuable experience during the seminar in Malang.

Malang, July 22, 2014

Editors

TABLE OF CONTENTS

MATHEMATICS EDUCATION:
THEORETICAL (CONCEPTUAL) ARTICLES

MATHEMATICS EDUCATION:
RESEARCH AND DEVELOPMENT ARTICLES

ISBN 978-602-71141-0-4

ISBN 978-602-71141-0-4

GRAPH THEORY

 ISBN 978-602-71141-0-4

DEVELOPING MST, TSP, AND VRP APPLICATION

Darmawan Satyananda
Department of Mathematics Education, State University of Malang
dsatyananda@gmail.com

Abstract

In course of Graph Theory Graf and Application of Graph Theory in Mathematics Department State University of Malang, students learn some algorithms for optimization problem, such as for searching Minimum Spanning Tree (MST), Travelling Salesman Problem (TSP), and Vehicle Routing Problem (VRP). For helping solving problems, for example for research purposes or Field Work (*Praktek Kerja Lapangan*), usually students use software for Graph application. Two of them are Grin and Giden. They have not been updated for a long time; meanwhile there are some new algorithms that are not contained in them. Algorithms studied in the course and are not contained in both software must be made as separate application. Skills developing the application is learned in Data Structure course, as discussed how structuring and operating data, with Graph and its application as one of the subject. An application is made to unify important algorithms studied in lectures and required in problem solving. The software implements algorithms grouped on problems of MST, TSP, and VRP. For data benchmark, TSPLIB dataset in XML format (Extensible Markup Language) is used.

Keywords: *Graph Application, TSPLIB, XML*

INTRODUCTION

In Mathematics Department, State University of Malang, Graph Theory subject is a mandatory course, and Graph Theory II (called Application of Graph Theory in Curriculum 2013 [hereinafter referred as AGT]) is an elective course. From the first time, interest of student who took AGT is quite high, approximately 30 of 60 students on each class.

AGT comprehends subject about application of several algorithms in Graph Theory for optimization problems, including finding of Minimum Spanning Tree (MST), Travelling Salesman Problem (TSP), Vehicle Routing Problem (VRP), Flow Problem, and Matching. In addition of studying the algorithm, there are assignments to the students in the form of real survey in company or institution to implement the specified algorithm according to the characteristics of the problem.

To date, students uses some software designed for the purpose, those are Giden (http://users.iems.northwestern.edu/~giden/), or Grin (http://www.oocities.org/pechv_ru/main.html). Some algorithms studied in PTG are contained in it, for instance *Nearest Neighbor* or MTVRP, however some recent algorithm cannot be found as the software has not been updated.

To overcome the obstacles, students are encouraged to create the application (computer program) that implements certain algorithm. This accommodated through Data Structure, which is electional course. The course discusses about how the representation of logical and physical data, as well as how the use of data in particular algorithm. One of the topics covered in this course (as contained in various textbook of Data Structure) is Graph. This because graph can represents a wide range of optimization problem, has various forms of data structure, and has a lot of algorithm (http://interactivepython.org/courselib/static/pythonds/Graphs/graphintro.html)

In the course of Data Structure in Mathematics Department UM, there are little adjustment in presentation of the material (as compared to the one listed in the textbook or taught in Computer Science department) because in here we more focused on how the implementation of algorithms in Graf Theory. The impact of the combination between Graph Theory II / Application of Graph Theory with Data Structure are produced of some theses about creation of computer program on the implementation of algorithms in Graph Theory, in the topic about MST, TSP, VRP, coloring, matching, and network flow. Some of the products are utilized by students doing Job Training (*Praktek Kerja Lapangan*), for example in optimizing of distribution of goods.

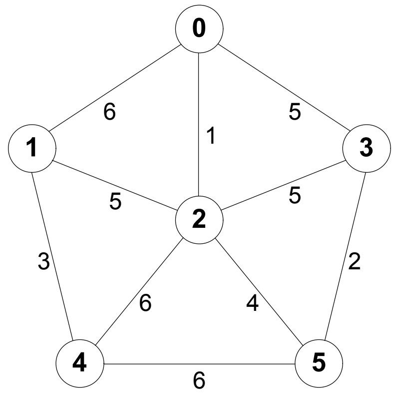
However, because generally thesis only discusses about an algorithm, it is required some software products if they want to compare or use multiple algorithms at once. This is what underlies the need of integrated application that can accommodate various needs of users.

LITERATURE REVIEW

Data structure of graph

A graph can be represented as an adjacency matrix that records connection of two vertices (forming an edge). The connection of two vertices can be stated with a certain value, such as distance of two vertices (Euclidean, Manhattan, or something else), or as edge weight. If there is no such connection, the value can be 0 or infinite. Symmetric TSP has square and symmetric adjacency matrix (value of row *i* and column *j* is equal to value of row *j* and column *i*), it means that an edge is recorded twice. In the other side, asymmetric TSP has square adjacency matrix but it is not a symmetric matrix. Another representation of graph is using *edge list* that only records graph edges. An edge is only recorded once (edge *uv* or *vu* only).

Figure 1 shows example of a graph and the representation of its data structure in the form of adjacency matrix and edge list.



(a)

	0	1	2	3	4	5
0	0	6	1	5	0	0
1	6	0	5	0	3	0
2	1	5	0	5	6	4
3	5	0	5	0	0	2
4	0	3	6	0	0	6
5	0	0	4	2	6	0

(b)

v1	0	0	0	1	1	2	2	2	3	4
v2	1	2	3	2	4	3	4	5	5	5
c	6	1	5	5	3	5	6	4	2	6
	0	1	2	3	4	5	6	7	8	9

(c)

Figure 1. (a) Example of a graph, (b) Graph representation as an adjacency matrix, (c) Graph representation as an edge list

Figure 1.b represents graph of 6 vertices as the adjacency matrix, there are 36 elements whereas half part of the matrix equals to another half part (above and below its main diagonal) because the graph is undirected (assume there are no multiple edges). Non-existing edge represented as element valued as 0. In figure 1.c, the list has 10 elements where each element contains information about the edge, and distance or weight of the edge. Number of list elements is as much the edge of the graph, and it is able to represent multiple edges

Some comparisons between usage of matrix and list (Satyananda, 2012):

 ISBN 978-602-71141-0-4

- Matrix is more suitable if edge searching or edge updating is often performed because the access of elements is done randomly, contrary to the list which is done sequentially (must traversing from the beginning of the list until the edge is found or not at all). Two vertices that form an edge are used as identifier of matrix' element to get the distance or weight of the edge.
- List is more suitable if the graph is not complete because there is no need to save unexisting edge as in the matrix with 0 or infinite value.
- List can be used to represent graph that has multiple edges between two vertices.

MST

Problem of MST is about how to find spanning tree in a graph (network), with least cost (Rosen, 2000). The spanning tree is a tree that connects all vertices in the graph. Some of the algorithms used in searching spanning tree are Kruskal (Rosen, 2000), Prim (Rosen, 2000), Boruvka (known also as Sollin algorithm) (Rosen, 2000), and Reverse-Delete (wikipedia).

TSP

TSP is the problem of modeling a salesman who departed from the place of origin, visited a number of other places exactly once, and must be returned to its original place, with the least possible cost or distance. When described, the route of the salesman will formed Hamilton cycle. For the purpose of this modeling, there are numerous algorithms that can be used, among of them are Nearest Neighbor Heuristic (Rosen, 2000), Nearest Insertion Heuristic (Rosen, 2000), Cheapest Link (Musser, 2007), and Genetic Algorithm (Mitchell, 1996, and Dwivedi, 2012).

VRP

In general, VRP problem is how to determine the route to distribute goods from the depot the customers, with the least cost / time / distance, or the smallest number of vehicle. In VRP there are one or more depot, one or more customer, one or more vehicle with specified capacity, distance / cost among depots and customers, service time on each customer, and time window as an overall time constraints that must be met (Rosen, 2000). Route generated by VRP could be more than one, depending on the constraints. VRP itself can be seen as set of TSP routes.

There are many variants of VRP in accordance with characteristic of problem. Kumar (2012) explained some of them:

a. CVRP (Capacitated Vehicle Routing Problem), where a vehicle with a limited capacity must get around to a number of customers who demand the goods in a certain amount.
b. VRPTW (Vehicle Routing Problem with Time Windows), where a vehicle has to serve a number of customers within a certain time limit at the least cost.
c. VRPBTW (Vehicle Routing Problem with Backhauls and Time Windows), in which the vehicle must deliver goods to the line-haul customer, while taking the goods from backhaul customers.

Benchmark data

Some source provides standardized test data used as a reference in a testing of the algorithm. One of them is TSPLIB that provides data for Symmetric TSP, Asymmetric TSP, Sequential Ordering Problem (SOP), and CVRP problems (http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/). The data from TSPLIB are in two kinds of format: TSPLIB standard format and XML format (to date, the XML format provided in TSPLIB website is only for TSP problem).

Here is sample for data in TSPLIB format (some data is removed, marked as ...).

```
NAME: berlin52
TYPE: TSP
COMMENT: 52 locations in
Berlin (Groetschel)
DIMENSION: 52
EDGE_WEIGHT_TYPE: EUC_2D
NODE_COORD_SECTION
```

```
1 565.0 575.0
2 25.0 185.0
...
51 1340.0 725.0
52 1740.0 245.0
EOF
```

In the file there are several descriptions, which are:

- NAME: identity of data instance
- TYPE: type of problem, in this case is TSP problem
- DIMENSION: number of vertices
- EDGE_WEIGHT_TYPE: type of edge weight, EUC_2D means Euclidean distance in 2-dimensonal plane (distance of 2 points). Other types are distance which is explicitly defined, Manhattan distance, and Geographical distance (Reinelt, 1995).
- NODE_COORD_SECTION: part shows data for every vertex 1 - *n*. If the TYPE is EUC_2D, then data for every vertex is coordinates of X-axis and Y-axis, and edge weight is calculated from the coordinates.

XML format is used as an alternative since it has more structured format than general format of TSPLIB, and it is easier to manage in Delphi or other recent programming language. In the original format of TSPLIB there are various representations of data (as an adjacency matrix, adjacency list, coordinate points to be calculated as a distance, directly as edge weight). In XML format there is only one type of data that is edge weight.

Here is sample of XML file for problem of TSP in the 5-vertices graph (some data is removed, marked as ...)

```
<?xml version="1.0"
encoding="UTF-8"
standalone="ignoredDigitsno"
?>
<travellingSalesmanProblemInst
ance>
 <name>Small 1</name>
 <source>Rostislav
Stanek</source>
 <description>A small
instance.</description>

<doublePrecision>15</doublePre
cision>

<ignoredDigits>5</ignoredDigit
s>
 <graph>
 <vertex>
 <edge
cost="2.000000000000000e+00">1
</edge>
 <edge
cost="1.000000000000000e+00">2
</edge>
 <edge
cost="2.000000000000000e+00">3
</edge>
 <edge
cost="1.000000000000000e+00">4
</edge>
 </vertex>
...
 <vertex>
 <edge
cost="1.000000000000000e+00">0
</edge>
 <edge
cost="2.000000000000000e+00">1
</edge>
 <edge
cost="1.000000000000000e+00">2
</edge>
 <edge
cost="2.000000000000000e+00">3
</edge>
 </vertex>
 </graph>
</travellingSalesmanProblemIns
tance>
```

In the data file, all edge weight states as *real* number, although the actual value is integer.

The advantage of XML format is that user can define their structure of the XML document and all kinds of information contained in it (in the form of tag definition, attribute, value, and data type). The structure of an XML document are described in XML Schema files (it has file extension named. xsd), which also serves to validator, to check the validity of an XML document. The XML document in a specific structure is valid

ISBN 978-602-71141-0-4

against a specific XML Schema file, so it will likely invalid when associated with different XML Schema file.

Delphi can generate unit XML data binding unit based on a specific XML Schema file for use in reading/writing XML documents that refer to the same XML Schema file (for XML document with different structure, need to be generated another XML data binding unit). This file is included in the main program as a Unit.

In the XML format above there are <graph> tags stating a graph. Inside are a number of <vertex> tags which symbolize one particular vertex. There are no attributes in this tag, so the vertex number implicitly indicated by the order of reading the file (starting from 0). Inside the <vertex> tag there are <edge> tags stating edge (arcs) formed by two vertices. This tag has "cost" attribute that states edge weight, and has a value that indicates vertex number that make up the edge of a particular <vertex>. When observed there is no edge between vertices with the same vertex number, it can be considered that the weight is 0.

For VRP case, the data from the VRP web (http://www.bernabe.dorronsoro.es/vrp/) can be used as standard data. Here is sample of the data file (some data are omitted, marked as ...)

```
2 4 50 4
0 80
0 80
0 80
0 80
 1 37 52 0   7 1 4 1 2 4 8
 2 49 49 0  30 1 4 1 2 4 8
...
49 48 28 0  18 1 4 1 2 4 8
50 56 37 0  10 1 4 1 2 4 8
51 20 20 0   0 0 0
52 30 40 0   0 0 0
53 50 30 0   0 0 0
54 60 50 0   0 0 0
```

According to the description in NEO website, the format comprises of:

- First line: VRP variant type, number of vehicle, number of customer, and the number of depot/day/vehicle
- Lines 2-(n+1) is time window of a route and capacity of each vehicle
- Lines (n+2) to (n+2+c) are for each customer, some of them are coordinate for each customer, demand, and service time
- Last d lines state data of each depot, particularly is position coordinate.

Not all data provided are used; it depends on the need or the problem.

METHOD

Application in general

Purpose of the application is to combine many algorithms in a package, in order to solve a problem in many ways to get the "best" result. There are three "groups" of algorithms to be implemented: MST, TSP, and VRP. Algorithms implemented in MST are Prim, Kruskal, Boruvka, and Reverse Delete algorithm. In TSP, algorithms implemented are Nearest Neighbor, Repetitive Nearest Neighbor, Nearest Insertion Heuristic, Cheapest Link, and *Genetic Algorithm*. Finally, in VRP algorithm of Multiple Depot Vehicle Routing Problem (MDVRP) is implemented.

Development steps

General steps in developing the application are:

a. Designing the application
   This step chooses algorithm to be implemented and its design. Requirement of the input and output is object to the algorithm and programming language used.
b. Implementing the design
   The design is implemented as an application (software) using Delphi programming language.
c. Testing the application
   Application testing is done using user-defined data and standardized data from TSPLIB, the VRP web, and other reputable source.

ISBN 978-602-71141-0-4

Designing the application

a. Characteristic of the application

The application has input in the kind of graph's vertices and edge weight. The iinput is performed from the application itself or from external data file. The graph is graphically visualized to help user understanding Output of the application is shortest path (for TSP problems), spanning tree (for MST problems), and distribution route (for VRP problems), in textual and graphical way.

b. Internal data structure

Data structure to represent graph basically is an adjacency matrix with Single-type element, to enable value of edge weight in floating point.

```
type
 MatriksBobot=class
 private
  matriks:array  of  array  of
Single;
...
end;
type Graph=class
 private
  mtBobot:MatriksBobot;
...
end;
```

On other cases that require finding edge which has least edge weight (edges are sorted by its weight) then structure in edge list is used as well.

```
type
 Sisi=class
  private
   u,v:SmallInt;
   bobot:Single;
...
end;
type ListSisi=class
  private
   lsSisi:TList;
   urut:Boolean;
...
end;
```

c. Interface of the application

User interface uses 3 types of component:

(1) Image area to enter vertices of the graph. Input of vertex is performed using mouse.

(2) Table (*grid*) sized *n* x *n* elements (*n*: number of vertex in a graph) to enter weight of edge *uv*. It is assumed that the graph is undirected, so that when value of line *u* column *v* (weight of edge *uv*) then value of line *v* column *u* will adjust. At the time a vertex is added to the image area, and then the number of rows / columns in the table is also increasing. Conversely when a value of edge weight is inputted then the value is shown in the graph in the image area. In the case of VRP, the grid is also used to enter data of customer requests

(3) *Text area* (*memo*) to display the result of calculation. The end result is also shown on the graph in different colors to indicate which side was selected as the spanning tree or Hamilton cycle or route of distribution.

The image area utilizes Image component and OnClick event to detect any click of mouse button in the image area. At the time click happens, then the coordinate position of the mouse pointer is recorded, and the number of rows/columns in a weight table (grid) increases. Vertex numbering starts at 0, but can be adjusted to start from 1 or by letter.

Weight table uses StringGrid component and onSetEditText event to detect any entry of edge weight. If the graph is undirected then value of weight entered to grid on row *u* column *v* will be copied to the row *v* column *u*. Weight of edges is not related to the actual distance between two vertices, but it is possible to automatically fill in the weights according to the distance between two vertices or as a random value. At the time an edge has weight then it is visualized on the image area.

Memo component is used to display textual results and other messages.

 ISBN 978-602-71141-0-4

d. Implementation of the algorithm

The algorithms implemented for the MST problem in this application are Kruskal, Prim, Boruvka/Sollin, and Reverse-Delete. For the TSP problem, Nearest Neighbor Heuristic, Repetitive Nearest Neighbor Heuristic, Nearest Insertion Heuristic, Cheapest Link, and the Genetic Algorithm are implemented. Repetitive Nearest Neighbor is Nearest Neighbor algorithm using all possible starting vertex. For problem of VRP, it is implemented MDVRP algorithm assisted by Clarke-Wright algorithm for route grouping.

e. External data

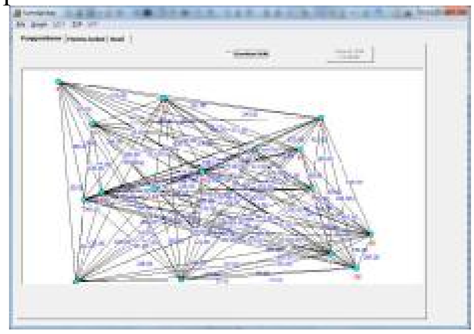
The graph in this application can be saved in a text file to be opened in any time. The information stored is the number of vertices, the coordinates of the vertex position in the image area, and the weight of each edge (if there is no edge between two vertices then the weight is 0).

Data input for TSP and MST problems using data from TSPLIB, and data files from the results of the previous manual input. The case was taken from TSPLIB gr17 (17 points) and gr120 (120 points).

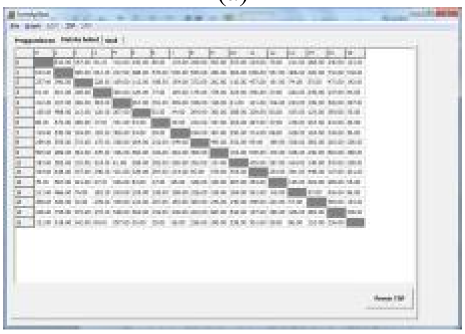
Data input to the VRP problem using the data from the VRP web, and data files from the results of the previous manual input. The case taken is p01 and p04. There are many variations MDVRP, so not all of the data available in the file is used. MDVRP algorithm in this application is using only the data: the number of depots, the number customers, demand of each customer, as well as the capacity of the vehicle. The number of vehicles is assumed to be enough (infinite and there is no time window and service time per customer).

RESULT

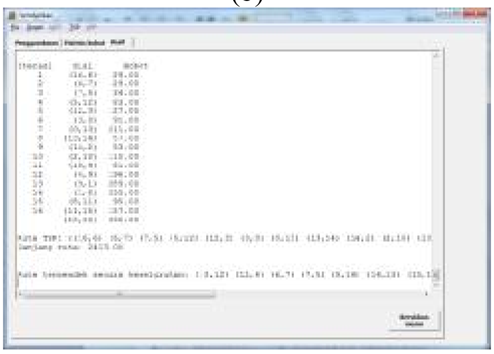
Here some screenshots of the application:



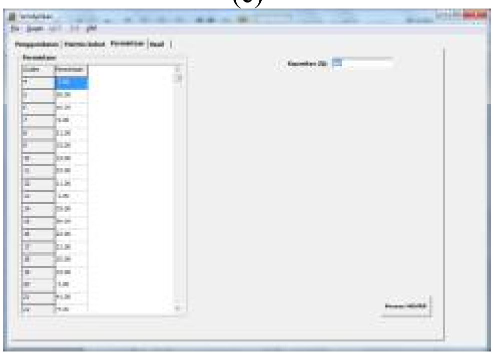
(a)



(b)



(c)



(d)

Figure 2. (a) Input of vertex data, (b) Weight table, (c) Output in textual form, (d) Table to fill customer demand on VRP problems

Result of the execution is displayed in Table 1 (routes are not shown).

MST		
Problem / Case	gr17	gr120
Kruskal	1421	5805
Prim	1421 (starting vertex: 0)	5805 (starting vertex: 0)
Boruvka	1421	5805
Reverse-Delete	1166	5303
TSP		
Problem / Case	gr17	gr120
Nearest Neighbor Heuristic	2187 (starting vertex: 0)	9351 (starting vertex: 0)
Repetitive Nearest Neighbor Heuristic	2178	8438
Nearest Insertion Heuristic	1870 (starting vertex: 0)	3120 (starting vertex: 0)
Cheapest Link	2189	8241
Genetic Algorithm	3586 (Generation: 3, Population: 30,. Crossover prob.: 0.7, Mutation prob.: 0.2, Elitism prob.: 0.2)	43895 (Generation: 3, Population: 150, Crossover Prob.: 0.7, Mutation Prob.: 0.2, Elitism Prob.: 0.2)
VRP		
Case	p01	p04
MDVRP	Depot 0: - Vehicle 0 : distance: 57.78, capacity: 48.00 - Vehicle 1 : distance: 51.17, capacity: 51.00 - Vehicle 2 : distance: 53.52, capacity: 56.00 - Vehicle 3 : distance: 26.96, capacity: 50.00 Depot: 1 - Vehicle 0 : distance: 91.09, capacity: 77.00 - Vehicle 1 : distance: 77.72, capacity: 80.00 - Vehicle 2 : distance: 39.75, capacity: 51.00 - Vehicle 3 : distance: 23.50, capacity: 54.00 Depot: 2 - Vehicle 0 : distance: 66.03, capacity: 79.00 - Vehicle 1 : distance: 38.36, capacity: 62.00 - Vehicle 2 : distance: 24.14, capacity: 36.00 Depot: 3 - Vehicle 0 : distance: 58.18, capacity: 74.00 - Vehicle 1 : distance: 57.40,	Depot: 0 - Vehicle 0 : distance: 114.64, capacity: 95.00 - Vehicle 1 : distance: 76.31, capacity: 100.00 - Vehicle 2 : distance: 89.79, capacity: 96.00 - Vehicle 3 : distance: 91.23, capacity: 96.00 - Vehicle 4 : distance: 71.91, capacity: 90.00 - Vehicle 5 : distance: 44.11, capacity: 94.00 - Vehicle 6 : distance: 46.27, capacity: 87.00 - Vehicle 7 : distance: 45.08, capacity: 97.00 - Vehicle 8 : distance: 14.14, capacity: 23.00 Depot: 1 - Vehicle 0 : distance: 115.62, capacity: 97.00 - Vehicle 1 : distance: 136.16, capacity: 94.00 - Vehicle 2 : distance: 59.63, capacity: 91.00 - Vehicle 3 : distance: 46.35, capacity: 59.00

 ISBN 978-602-71141-0-4

	capacity: 59.00 Total: 665.6	- Vehicle 4 : distance: 49.10, capacity: 93.00 - Vehicle 5 : distance: 59.72, capacity: 98.00 - Vehicle 6 : distance: 75.58, capacity: 100.00 - Vehicle 7 : distance: 57.57, capacity: 48.00 Total: 1193.21

Comparing to the best result from each case, Table 2 shows the comparison:

Table 2. The recorded best results

Problem / case	Best result
TSP (https://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/STSP.html)	
gr17	2085
gr120	6942
MDVRP (http://www.bernabe.dorronsoro.es/vrp/)	
p01	576.87
p04	1001.59

The best data for MST is not obtained because there is no record that provides the best results for the dataset from TSPLIB.

In general, the empirical results from the execution of the application against the test data is:

(a) In the TSP problem of case gr17, three algorithms (Nearest Neighbor, Repetitive Nearest Neighbor, Cheapest Link) give results that do not differ much, although the result was not any better than record in Table 2. Nearest Insertion Heuristic algorithm gives best result, while the Genetic Algorithm gives the worst result

(b) In the TSP problem of case gr120, only two algorithms (Repetitive Nearest Neighbor and Cheapest Link) provide a worse result but not too much different, followed by Nearest Neighbor, and a Genetic Algorithm that gives the worst results. Nearest Insertion algorithm gives the best results.

(c) Genetic Algorithm gives the worst result, even tend to differ very much as number of vertex increases, presumably because it involves a random process at the time of the initial determination of the chromosome so that the results obtained can be uncertain. For TSP problem in graph with many vertices, Genetic Algorithm is discouraged because deviations can be even greater.

(d) In MDVRP problems, for both cases the results obtained are relatively closer to the best results, although with the larger value. Possible causes are due to differences in the determination of the customer bounded to a particular depot (clustering step) and the time saving for the determination of the initial route.

CLOSING

Applications created is a stub for the development of applications that can integrate various algorithms necessary and commonly used. The results still have not been able to approach the best results from standard data, but the differences that occur can be tolerated. The better algorithm is still searched, to get a more better result.

Suggestions for the development of such applications are:

(a) Searching algorithm which can provide relatively better results

(b) Improving the implementation of the algorithm in order to obtain a more efficient process

(c) Development of an application dinamically, in order to ease integration of a new algorithm into the application.

REFERENCES

Dwivedi, Varshika, et.al. 2012. *Travelling Salesman Problem using Genetic Algorithm*. Proceeding of National Conference on Development of Reliable Information Systems, Techniques and Related Issues (DRISTI) 2012 (http://research.ijcaonline.org/dristi/number1/dristi1007.pdf, accessed in 1 Mei 2014)

Kumar, Suresh Nanda, and Ramasamy Panneerselvam. 2012. *A Survey on the Vehicle Routing Problem and Its Variants*. Intelligent Information Management, 2012, ed. 4, p. 66-74 (www.scirp.org/journal/PaperDownload.aspx?PaperID=19355, accessed in 20 Mei 2014)

Masruroh, Annisa. 2012. *Algoritma Clark and Wright Pada Multi-Depot Vehicle Routing Problem (MDVRP)*. Universitas Negeri Malang, unpublished theses

Mitchell, Melanie. 1996. *An Introduction to Genetic Algorithm*. MIT Press.

Musser, Gary, dan Lynn Trimpe. 2007. *A Mathematical View of Our World*. Cengage Learning (http://books.google.co.id/books?id=hplkLvGkuzwC&pg=PA389&dq=cheapest+link+algorithm&hl=en&sa=X&ei=PyVdU8f3IZGOiAfR-4HoBg&ved=0CC0Q6AEwAA#v=onepage&q=cheapest%20link%20algorithm&f=false, accessed in 1 Mei 2014)

Reinelt, Gerhard. 1995. *TSPLIB 95*. (http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/DOC.PS, accessed in 1 Mei 2014)

Rosen, Kenneth H, et.al (editor). 2000. *Handbook Of Discrete And Combinatorial Mathematics*. Florida: CRC Press.

Satynananda, Darmawan. 2012. *Struktur Data*. Universitas Negeri Malang, unpublished dictates

_____. *TSPLIB-XML Description* (http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/XML-TSPLIB/Description.pdf, accessed in 1 Mei 2014)

_____. *Reverse-Delete Algorithm* (http://en.wikipedia.org/wiki/Reverse-delete_algorithm, accessed in 1 Mei 2014)

ISBN 978-602-71141-0-4