Assignment 3.1: Incorporation of graph-specific information to solve the minimum dominating set problem using a genetic algorithm

L. Beyers
Department of Applied Mathematics
Stellenbosch University
21591644
E-mail: 21591644@sun.ac.za

***Abstract*—The Minimum Dominating Set (MDS) problem aims to find the minimum dominating set of a graph G. The solution space of the MDS problem is, however, highly dependent on the nature of G. This report investigates the effect of using information about G to more effectively apply a genetic algorithm to solve the MDS problem. The coverage and size of solutions are considered to determine whether implementations which use graph-specific information are more successful. It is found that statistically significant differences in performance occur for sparse graphs and implementation behaviour is not greatly affected by the use of graph-specific information. The use of graph-specific information therefore may improve performance of a genetic algorithm on the MDS problem, though further study is required.**

I. Introduction

Graph domination was first considered by Claude Berge in 1958 [6]. The Minimum Dominating Set (MDS) problem concerns itself with finding the smallest possible set of vertices which dominate a given graph. The MDS problem is NP-hard, which is why the use of a meta-heuristic to solve it is appropriate. It can be difficult to prove optimality of a found solution and so the fitness of a solution should be measured relatively.

The use of a genetic algorithm to find solutions to the MDS problem may provide adequate results, but current implementations do not use information about the given graph [3]. In graph domination, it is possible to use information about the given graph to make computationally inexpensive good guesses as to which vertices are likely to contribute to good solutions. The aim of this report is to use a general scheme to allow the meta-heuristic, in this case a genetic algorithm, to use graph-specific information. This report aims to answer the question of whether this incorporation of problem-specific information improves the solution-finding ability of the meta-heuristic. The information which is incorporated takes on the form of a static desirability score.

II. Background

Some context is given in this section which helps motivate the choices that are made for the purpose of this report.

Basic graph theoretic concepts are presented in subsection II-A. The nature of the MDS problem is discussed in subsection II-B. Current approaches to the MDS problem are discussed in subsection II-C.

A. Graph theoretic background

Some graph theory knowledge is required to understand the nature of the dominating set problem.

Let a graph $G = (V, E)$ contain a set of vertices and a set of edges. A vertex $v_1 \in V$ is *adjacent* to a vertex $v_2 \in V$ if $(v_1, v_2) \in E$. The *open neighbourhood* (referred to hereafter as the *neighbourhood*) of a vertex v is the set of all vertices to which v is adjacent. A vertex v *dominates* itself as well as every vertex in its neighbourhood. A *dominating set* of G is a subset $V' \subseteq V$ for which every vertex in V is dominated by at least one vertex in V'. A *minimum dominating set* is a dominating set which contains the minimum possible number of vertices required to dominate G. This minimum possible number of vertices is known as the *domination number* of G, denoted by $\gamma(G)$. [4]

B. Nature of the minimum dominating set problem

The characteristics of the solution space of the MDS problem are highly dependent on the graph G which is being considered. Changes in the cardinalities of the vertex and edge sets of G have unpredictable effects on the solution space. Even when the the cardinalities of the vertex and edge sets of G remain constant, changes to graph structure affect the solution space of the MDS problem in unpredictable ways [4]. This dependency poses a challenge when attempts are made to generalise approaches to finding solutions. The use of a meta-heuristic which adapts to the search space is therefore a good approach to finding solutions. This is why a genetic algorithm is appropriate for the MDS problem.

There are some general indicators for good solutions which can be incorporated into search algorithms for the MDS problem. Vertex-specific indicators which use local information are ideal since they only utilise small amounts of accessible information. This idea motivates the design of the desirability scores outlined in section III-D.

C. Current approaches

The MDS problem can be approached using theoretic results or using nondeterministic approaches. The former is limited in scope and the latter is limited in optimality. If an approach like a heuristic exploits the theoretical knowledge we have about the MDS problem, it has the potential to be more efficient and to obtain better solutions than it would otherwise.

One approach that is considered in this report is that of A. Hedar et al.

III. Methodology

The implementation of a strategy to answer the research question is considered in this section.

The representation of solutions is discussed in subsection III-A. A fitness evaluation approach is introduced in subsection III-B. The method used to rank solutions is discussed in subsection III-C. A desirability score framework is introduced in subsection III-D. Mutation is discussed in subsection III-E and crossover in subsection III-G.

A. Solution representation

For a graph G, a feasible solution to the domination problem is a set $V' \subset V$ which dominates G. This report also considers infeasible solutions which cover only a proper subset of V. In the implementation investigated in this report, a solution is represented by a binary string $\mathbf{x}_i$ of length $|V|$. For every vertex $v_j \in V$, if $\mathbf{x}_i$ contains v, then $x_{ij} = 1$. Else $x_{ij} = 0$. Each bit x_{ij} in the solution is referred to as a *gene*.

B. Two-factor fitness evaluation

The fitness of a solution $\mathbf{x}_i$ representing a set of vertices $V' \subseteq V$ is evaluated in terms of the cardinality of V' and the cardinality of the vertex set dominated by V'. Let V^+ be the set of vertices dominated by V'.

$$fitness = \frac{|V^+|}{|V'|} \tag{1}$$

Eq. 1 is adjusted in order to favour solutions which dominate G. This adjustment is scaled by the standard deviation of the fitness values obtained in Eq. 1. It is implemented as follows:

for $\mathbf{x}_j$ in $population$ **do**
  $f_j = \frac{|V^+|}{|V'|}$
end for
$\sigma = \sigma(\mathbf{f})$
for $\mathbf{x}_j$ in $population$ **do**
  if $\mathbf{x}_j$ dominates G **then**
    $f_j = f_j + \lambda_3\sigma$
  end if
end for

In the implementation considered in this report, λ_3 is chosen to be three. Let us assume that the scores are normally distributed.

Before scores are adjusted, dominating solutions are likely to have lower scores than non-dominating solutions. This is because a small change in the size of the solution set effects a large change in the score value, since the solution set is typically much smaller than the set of vertices it dominates. Non-dominating solutions are likely to have smaller solution sets and so, their scores are likely to be higher than those of dominating solutions.

The choice of $\lambda_3 = 3$ ensures that the dominating solutions are given an advantage of three standard deviations over the non-dominating solutions. Because non-dominating solutions are likely to have higher scores before the adjustment, this shift can be expected to result in overlap of solution score ranges. The shift also introduces a very high likelihood that the highest score will be assigned to a dominating set. [5]

More work can be done on understanding the distribution of these scores.

C. Ranking solutions

Solutions are ranked based on their fitness values. The rank of a solution is used in the replacement strategy and in the parent selection operator. The use of rank instead of fitness in the desirability score update is to lower the rate at which desirability stagnates.

D. Desirability scores

1) Desirability representation: Each vertex v_j in G is assigned a desirability score d_j. This score is a measure of the contribution of v_j to a dominating set. d_j is initialised based on the structure of G. Static desirability scores keep this initial value for the duration of the execution of the algorithm, while dynamic desirability scores are updated according to the performance of solutions containing v_j. This report only considers static desirability.

2) Initialisation: The initialisation of desirability d_j of vertex v_j considers the degree of v_j as well as the average degree of the neighborhood of v_j. A vertex v_j with a high degree is desirable in terms of domination, since the subgraph dominated by v_j is large. However, if v_j is adjacent to many other vertices of the same degree or higher, it may be more desirable that its neighbour be in the proposed dominating set. In this case, the desirability of v_j is decreased.

First, an assessment of the structural desirability qualities of vertex v_j with neighbourhood N_j is performed, with values stored as $d_j(-1)$:

$$d_j(-1) = |N_j| - \frac{1}{|N_j|}\sum deg(v_k),\ v_k \in B_j.$$

For randomly generated graphs, it is conjectured that the set of $d_j(-1), j = 1 : n$ form a normal distribution with an expectation of 0. For a small graph with fewer than 100 vertices, there may be errors in calculating the standard deviation of $d_j(-1)$. The range of $\mathbf{d}(-1)$ is therefore substituted for standard deviation as a measure of spread. The scores are scaled according to their range as follows:

$$d_j(0) = \frac{d_j(-1)}{\lambda_1(\max\limits_{k=1:n}(d_k) - \min\limits_{k=1:n}(d_k))} - \lambda_2.$$

Parameter λ_1 is a scaling factor which controls the size of the effect of the desirability qualities on the desirability scores. It

is chosen to be 2 on the basis that 99.5% of data on a normal distribution falls within 2 standard deviations from the mean. [5]

This implies that, after scaling, roughly 99.5% of the desirability scores will lie in the range $(-0.5, 0.5)$.

Parameter λ_2 is chosen to be 0.5 based on the assumption that, for a randomly generated graph, the expectation of $d_j(-1)$ is 0. [5] Centering the scores about 0.5 instead ensures that they are in a reasonable range to be used as a probability.

3) Uses: Desirability is vertex-specific information, but when represented in list format as $\mathbf{d} = [d_1, d_2, \ldots, d_j]$, desirability should be close to good solutions.

The desirability scores are used for tournament selection of the parents. The potential parents in the tournament are then evaluated based on desirability instead of fitness. The potential parent which is most similar to $\mathbf{d}$ wins the tournament.

E. Mutation

The mutation rate of the genetic algorithm decreases with time and is dependent on the iteration number of the genetic algorithm. The mutation rate is calculated as

$$p_m(t) = 1240 + \frac{0.11375}{2^t}$$

as suggested by Fogarty [7]. It is not necessary to correct for binary encoding since the binary solutions are not representative of a number.

F. Parent selection

Each individual in the population is a parent at least once per generation. This assigned parent is referred to as the *current parent*. A second parent is selected using tournament selection. The tournament is held between two randomly selected individuals. The two possible measures by which potential parents can win is closeness to $\mathbf{d}$ or rank. Tournament selection with each of these measures is implemented.

G. Crossover

The crossover operator used is one-point crossover as presented by Engelbrecht [2]. Crossover is executed at a crossover rate of $c_r = 0.5$. Each crossover generates one offspring only.

H. Replacement strategy

Offspring replace the current parent (not necessarily the worst parent) if they rank better than the current parent. If the rank of the offspring and the current parent are equal, the offspring replaces the parent with a probability of $p_r = 0.5$.

IV. Empirical procedure

The experiment was carried out under conditions which are detailed in this section. The success of each implementation was measured according to criteria detailed in this section.

Benchmark problems are discussed in subsection IV-A. Algorithm implementation is discussed in subsection IV-B. Algorithm behaviour indicators are discussed in subsection IV-C.

A. Benchmark problems

Random graphs were created for the use of this experiment. These graphs vary in the sizes of their edge and vertex sets. The focus of this report is on small graphs. Random graphs $G_i = (V, E)$ with $n = |V|$ and a probability of any two vertices being connected of p_e were generated and stored. Values for n and p_e are given in Table I.

TABLE I
GRAPHS USED AND THEIR PROPERTIES

G_i	n	p_e	$\lvert E\rvert$
G_1	10	0.5	27
G_2	20	0.5	95
G_3	50	0.5	613
G_4	100	0.5	2457
G_5	10	0.4	25
G_6	20	0.4	93
G_7	50	0.4	504
G_8	100	0.4	1901
G_9	10	0.3	14
G_{10}	20	0.3	54
G_{11}	50	0.3	395
G_{12}	100	0.3	1491
G_{13}	10	0.2	15
G_{14}	20	0.2	38
G_{15}	50	0.2	228
G_{16}	100	0.2	941
G_{17}	10	0.1	6
G_{18}	20	0.1	24
G_{19}	50	0.1	132
G_{20}	100	0.1	506

B. Algorithm parameters

The genetic algorithm is implemented with a population size of $p = 20$. Variation of population size is left to future studies. The genetic algorithm is run 10 times per graph using desirability scores for parent selection and 10 times per graph using rank for parent selection. Each independent run continues for 3000 generations.

C. Algorithm behaviour indicators

Three measures are used to analyse behaviour of the algorithm: the average size of the set dominated by a solution in any given generation, d_{si}, the average size of a solution in any given generation s_{si} and the size of the best solution obtained in the final generation s_f.

The average size of the set dominated by a solution in any given generation is an indicator of how well the solutions in the generation cover the graph.

The average size of a solution in a generation is the main measurement of solution improvement throughout generations because the size of feasible solutions is the parameter which must be optimised for an optimal solution to the MDS problem. Infeasible solutions are taken into account in this measure, because their effect can be monitored and accounted for using d_{si}.

The best solution obtained in the final generation is the final result produced by the algorithm. It is a simple measure of performance.

D. Statistical analysis

The Mann-Whitney U two-tailed test was used to determine whether there is a statistically significant difference between final results obtained by the algorithm with and without the use of desirability scores.

The research hypothesis states that the use of desirability scores will increase performance. If there is a statistically significant difference between the performance of the algorithm when desirability is used versus when it is not, then further investigation will be required.

There were 7 independent runs per graph. Two sets of 7 values per graph were compared. These values represent the sizes of the final solutions found.

For a two-tailed test, we consider the null hypothesis, H_0, which states that the two sets of data considered are sampled from the same underlying distribution. The level of significance was chosen to be $\alpha = 0.05$. With these parameters, the critical value is 8 [8]. Therefore, if the Mann-Whitney U result for any graph is less than or equal to 8, then the null hypothesis is rejected and there is a statistically significant difference between the two sets of values.

Similarly for a one-tailed test, let the hypothesis be that the sizes of solutions found by the algorithm using desirability scores were smaller than those found without using desirability. Once again, there are 7 final solutions found per graph and the level of significance is chosen to be $\alpha = 0.05$. The critical value is then 11 [8]. If Mann-Whitney U result for any graph is less than or equal to 11, then there is no statistical significance between the two sets of values.

V. Research results

Let the implementation of the genetic algorithm using rank for parent selection be denoted by I_r. Let the implementation of the genetic algorithm using desirability for parent selection be denoted by I_d.

The differences in implementation behaviour are discussed in subsection V-A. Statistical significance of implementation performance is discussed in subsection V-B.

A. Implementation behaviour

The size of the set of vertices dominated by each solution was measured at each generation. Additionally, the size of each solution was measured at each generation. Figure 1 depicts these measures for each generation, averaged over the solutions and the 7 independent runs.

Dashed lines, marked with 'des' in the legend, represent measures taken for I_d. Straight lines represent measures taken for I_r.

Figure $1e)$ illustrates that the average sizes of solutions found by I_d and I_r undergo similar changes over generations for 10-vertex graphs. The greatest difference occurs for G_{17}, which has the smallest number of edges. The solution set size becomes stable at generation 800 for I_d, while the solution set size becomes stable at generation 300 for I_r.

Figure $1a)$ illustrates that generations 800 and 300 are where the respective solution sets begin to dominate all of G_{17}.

This difference will affect the computational expense of finding solutions, but since G_{17} only has 10 vertices, this is not a significant drawback.

Figure $1f)$ illustrates that the average sizes of solutions found by I_d and I_r undergo similar changes over generations for 20-vertex graphs. This similarity is reflected in Figures $1g)$ and $1h)$ for 50-vertex graphs and 100-vertex graphs, respectively.

Though the implementations I_d and I_r appear to behave similarly according to Figures $1e)$ - $1f)$, it can be noted that average solution size appears to be more erratic for I_d with sparse graphs. This can be expected, since the desirability model used in I_d is a static measure which does not necessarily encourage search in the same direction as the scoring and ranking system.

Figures $1b)$ - $1d)$ confirm that there is greater difference in behaviour of I_d and I_r when sparse graphs are considered.

An anomaly is illustrated by Figure $1b)$, where it takes 150 more generations for the average size of the set of dominated vertices of G_{18} for I_r to stabilise than for those of I_d to stabilise. The average solution sizes of G_{18} are similar for I_d and I_r for generations 0 to 500 and so this dip in performance is not reflected in the average solution sizes for I_r.

B. Statistical analysis of implementation performance

The Mann-Whitney U test was performed on the sizes of the best solutions found per graph per run. Results from the Mann-Whitney U test are displayed in Table II. There is statistically significant evidence at $\alpha = 0.05$ that the null hypothesis can be rejected for graphs G_9 to G_{20}. The performance of the algorithm for graphs with a lower edge density was therefore significantly altered by the use of desirability scores in the parent selection process. It cannot be said from Table II that performance differed significantly for graphs G_1 to G_8, since the results are not statistically significant.

VI. Conclusion

The aim of this report was to discover whether the use of graph-specific information to guide parent selection would improve the performance of a genetic algorithm on the Minimum Dominating Set (MDS) problem for small graphs. Behaviour of the implementation which utilised graph-specific information appeared similar to that of the algorithm which did not utilise graph-specific behaviour. This is significant because the graph-specific information was used as a major driver for exploitation of the gene pool and even though it was a static measure, it produced similar behaviour to that of an implementation which used dynamic, relative scores to drive exploitation of the gene pool. The desirability score introduced in this report is therefore a reliable measure of the closeness of a solution to a good solution. It was further discovered that performance of the algorithm was significantly affected by the inclusion of graph-specific information for sparse graphs.

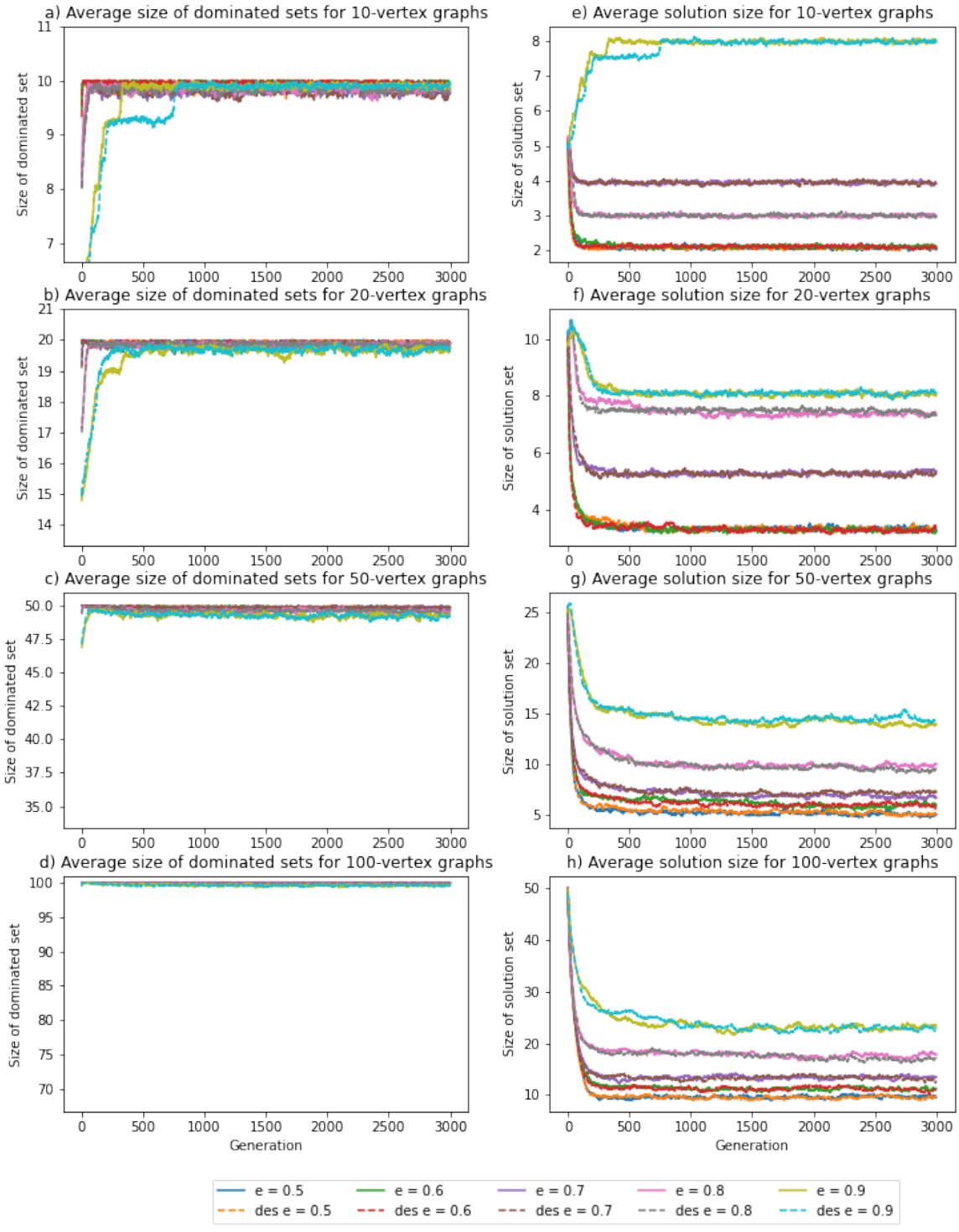

Fig. 1. Sizes of dominated sets and solutions for graphs with different numbers of vertices.

TABLE II
MANN-WHITNEY U RESULTS FOR THE TWO-TAILED TEST

G_i	*U*	*p*	*Status of null hypothesis*
G_1	19.5	0.24	Not rejected
G_2	21.5	0.33	Not rejected
G_3	24	0.50	Not rejected
G_4	23.5	0.47	Not rejected
G_5	16.5	0.14	Not rejected
G_6	23.5	0.47	Not rejected
G_7	20	0.30	Not rejected
G_8	20	0.30	Not rejected
G_9	0	0	Rejected
G_{10}	0	0	Rejected
G_{11}	0	0	Rejected
G_{12}	0	0	Rejected
G_{13}	0	0	Rejected
G_{14}	0	0	Rejected
G_{15}	0	0	Rejected
G_{16}	0	0	Rejected
G_{17}	0	0	Rejected
G_{18}	0	0	Rejected
G_{19}	0	0	Rejected
G_{20}	0	0	Rejected

Further statistical analysis is required to determine whether the effect is improvement.

Future work may include the use of the desirability scores in other selection and decision processes of the genetic algorithm as well as further study into the effects of using graph-specific information when working with larger graphs.

REFERENCES

[1] L. Sanchis, "Experimental analysis of heuristic algorithms for the dominating set problem", in *Algorithmica*, 33(1):pp.3–18, 2002.

[2] A. Engelbrecht, "Computational Intelligence: An Introduction", Second Edition, John Wiley & Sons Ltd, 2007.

[3] A. Hedar, R. Ismail, G. El Sayed, K. Khayyat, "Two Meta-Heuristics for the Minimum Connected Dominating Set Problem with an Application in Wireless Networks", in *3rd International Conference on Applied Computing and Information Technology/2nd International Conference on Computational Science and Intelligence (ACIT-CSI)*, pp. 355–362, 2015.

[4] G. Chartrand, P. Zhang, "A first course in graph theory", Dover Publications, Inc. , pp.361-370, 2012.

[5] S. Glen, "Empirical Rule (68-95-99.7) & Empirical Research", from *StatisticsHowTo.com: Elementary Statistics for the rest of us!*, available at *https://www.statisticshowto.com/probability-and-statistics/statistics-definitions/empirical-rule-2/*, 2022.

[6] C. Berge, "Théorie des Graphes et Ses Applications", Dunod, Paris, 1958.

[7] T.C. Fogarty, "Varying the Probability of Mutation in the Genetic Algorithm", in *J.D. Schaffer, editor, Proceedings of the Third International Conference on Genetic Algorithms*, pp. 104–109, San Mateo, C.A., 1989.

[8] W. LaMorte, "Mann Whitney U Test (Wilcoxon Rank Sum Test)", available at *https://sphweb.bumc.bu.edu/otlt/mph-modules/bs/bs704_nonparametric/bs704_nonparametric4.html*, Boston University School of Public Health, 2017.