World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:3, No:5, 2009

Multi-objective Optimization of Graph Partitioning using Genetic Algorithm

M. Farshbaf, and M. R. Feizi-Derakhshi

Open Science Index, Computer and Information Engineering Vol:3, No:5, 2009 publications.waset.org/11319/pdf

***Abstract*—**Graph partitioning is a NP-hard problem with multiple conflicting objectives. The graph partitioning should minimize the inter-partition relationship while maximizing the intra-partition relationship. Furthermore, the partition load should be evenly distributed over the respective partitions. Therefore this is a multi-objective optimization problem (MOO). One of the approaches to MOO is Pareto optimization which has been used in this paper. The proposed methods of this paper used to improve the performance are injecting best solutions of previous runs into the first generation of next runs and also storing the non-dominated set of previous generations to combine with later generation's non-dominated set. These improvements prevent the GA from getting stuck in the local optima and increase the probability of finding more optimal solutions. Finally, a simulation research is carried out to investigate the effectiveness of the proposed algorithm. The simulation results confirm the effectiveness of the proposed method.

***Keywords*—**Graph partitioning, Genetic algorithm, Multi-objective optimization, Pareto front.

I. Introduction

NOWADAYS, Graph partitioning is used directly in a wide range of problems. Graph partitioning involves dividing a set of objects into a specified number of partitions according to the minimization of some optimization criterion additive over the partitions. The technique used here presents how to distribute N object in M partitions such that the intra-partition relationship is maximized and inter-partition relationship is minimized [1] and simultaneously make the partition load be evenly distributed over the respective partitions which sets constraint for the optimization problem. So the objective functions are more than one and conflicting with each other. One of the approaches to Multiple-Objective Optimization (MOO) according to Pareto optimization is NSGA-II. In this paper NSGA-II has been used. Proposed improvements are injecting best solutions of last runs into the first generation of the next runs and also storing the non-dominated set of the previous generations to combine with the last generation's non-dominated set. These improvement methods prevent the GA from getting stuck in the local optima, also increase the opportunity for better solutions to mate and to participate in the generation of last Pareto front set.

M. Farshbaf is with Tabriz University, Tabriz, Iran (e-mail: farshbaff@ yahoo.com).

M. R. Feizi-Derakhshi is with the Department of Computer, Tabriz University, Tabriz, Iran (e-mail: mfeizi@tabrizu.ac.ir).

The paper is organized as follows: Section II briefly introduces the multi-objective optimization methods and also NSGA-II idea. In section III graph partitioning is modeled as a multi-objective optimization problem. GA implementation to solve this multi-objective optimization problem is discussed in section IV and effectiveness of the proposed method is investigated through a simulation example. The paper is concluded in Section V.

II. Brief Overview of Multiple Objective Optimization

In many applications the cost function has multiple, often times conflicting, objectives. There are two general approaches to multiple-objective Optimization: weighted sum method and Pareto-based Approach [2]. The weighted sum method aggregates individual objective functions into a single composite function, but the problem lies in the proper selection of the weights to characterize the decision-maker's preferences. In practice, it can be very difficult to precisely and accurately select these weights, even for someone familiar with the problem domain. Compounding this drawback is that scaling amongst objectives is needed and small perturbations in the weights can sometimes lead to quite different solutions [3]. In order to overcome such difficulties, Pareto-based evolutionary optimization has become an alternative to classical techniques such as weighted sum method. This approach was first proposed by Goldberg in [4].

A. Pareto-based Approach

Solutions to a multi-objective optimization problem can be expressed mathematically in terms of non-dominated points, i.e., a solution is dominant over another only if it has superior performance in all criteria. A solution is said to be Pareto-optimal if it can not be dominated by any other solution available in the search space. While moving from one Pareto solution to another, there is always a certain amount of sacrifice in one objective(s) to achieve a certain amount of gain in the other(s). Pareto optimal solution sets are often preferred to single solutions because they can be practical when considering real-life problems.

The NSGA-II is a very robust multi-objective optimization algorithm based on Pareto method. The idea behind NSGA is that a ranking selection method is used to emphasize good points and a niche method is used to maintain stable subpopulations of good points. Before the selection is performed, the population is ranked on the basis of an individual's non-domination. The non-dominated individuals present in the population are first identified from the current

Open Science Index, Computer and Information Engineering Vol:3, No:5, 2009 publications.waset.org/11319/pdf

population. Then, all these individuals are assumed to constitute the first non-dominated front in the population and assigned a large dummy fitness value. The same fitness value is assigned to give an equal reproductive potential to all these non-dominated individuals. Then, these individuals are removed from the population. Next all the non-dominated individuals of this smaller population are found and assigned a rank of two. This process continues until all the individuals are assigned a rank and fitness value. Then non-dominated solutions have more chance to be selected for mating because they have better fitness than other solutions. The main advantage of this approach is that a global map of multiple objectives to a singe objective function using a non-dominated sorting procedure. Details on NSGA-II can be found in [5].

The algorithm proposed in this paper is an extension to NSGA-II in graph partitioning problem, but the modifications are general and could be applied to other MOEAs. A database of best solutions has been used to save last Pareto set of various runs. Due to high cost of generating elitist individuals, solutions of the database are injected into the first generation of next runs to have some elitist solutions in the beginning. Also Pareto optimal set of each generation are being stored to combine with the last generation's non-dominated set.

III. Modeling of the Optimization Problem

The proposed methodology has been tested for a real partitioning system with dimensions that are significantly larger than the ones frequently found in the literature.

A. Model Description

Graph partitioning in general follows three principles [6]:
(1) Divide those objects having closer relationships into the same partition, such that
Intra-partition relationship is maximized.
(2) Inter-partition relationships should be minimized.
(3) Balance the partition load over the resultant partitions,
Such that the total load burden is evenly distributed in respective partitions. This principle prevents the partitioning performance from being significantly influenced if new elements are added to an arbitrary partition.

The main basis for the graph partitioning problem is the relationship between arbitrary two nodes, which can be described by a relationship matrix. The relationship (R) between arbitrary two nodes requires careful consideration of common and uncommon grounds between two nodes A and B and can be measured as:

$$R(A,B) = \frac{N(A \cap B)}{N(A \cup B)}. \quad (1)$$

$N(A \cap B)$ in Eq. (1) considers common grounds, and $N(A \cup B)$ considers both common and uncommon grounds between two nodes.

If i = {1, 2, 3, … , n} is used to represent the nodes, then the relationship matrix can be constructed as following:

$$R = \begin{bmatrix} 0 & r_{12} & \dots & r_{1n} \\ r_{21} & 0 & \dots & r_{2n} \\ \dots & \dots & \dots & \dots \\ r_{n1} & r_{n2} & \dots & 0 \end{bmatrix}. \quad (2)$$

Where the element r_{ij} represents the relationship between node i and node j which is computed from Eq. (1). Nodes do not have relationship with themselves, so diagonal elements in the matrix are all zeros.

B. Problem Definition

Use k = {1, 2, 3, …, K} to represent the partition, and define

$$x_{ik} = \begin{cases} 1, & i \in Partition\ \ k, \\ 0, & i \notin Partition\ \ k. \end{cases} \quad (3)$$

For any partition, $x_{ik}(1 - x_{jk})$ is equal to 0 if both node i and node j belong to partition k, and is equal to 1 if node i belongs to partition k while j belongs to a different partition. Consequently the aggregate inter-partition relationship among all partitions can be formulated as

$$C = \sum_{k=1}^{K} \sum_{j=1}^{n} \sum_{i=1, i \neq j}^{n} x_{ik}\left(1 - x_{jk}\right) r_{ij}. \quad (4)$$

The intra-partition relationship of any partition can be formulated as

$$w(k) = \sum_{j=1}^{n} \sum_{i=1}^{n} x_{ik} x_{jk} r_{ij}. \quad (5)$$

And the aggregate intra-partition relationship among all partitions can be formulated as

$$W = \sum_{k=1}^{K} \sum_{j=1}^{n} \sum_{i=1}^{n} x_{ik} x_{jk} r_{ij}. \quad (6)$$

Remark. In our data set, always $r_{ij} = r_{ji}$, therefore the algorithm only uses the elements above or below the diagonal.

The maximum partition load imbalance among all partitions can be defined by the maximum partition load difference

$$I = \max_{1 \le i, j \le K} \left| w(i) - w(j) \right|. \quad (7)$$

Three graph partitioning principles are equivalent to minimize Eqs. (4) and (7) and maximize Eq. (6) simultaneously, which is multi-objective optimization problem. In the proposed algorithm Eqs. (4) and (6) are considered as optimization objectives and Eq. (7) is moved to the constraint set that must be established for each of these former objectives.

Open Science Index, Computer and Information Engineering Vol:3, No:5, 2009 publications.waset.org/11319/pdf

IV. Implementation of the GA

Genetic Algorithms were developed by John Holland at the University of Michigan. They are search algorithms based on the mechanics of natural selection and natural genetics [7]. The algorithm begins with an initial solutions population of the problem. This population is generated randomly. Each one of these solutions must be evaluated by means of a fitness function; the result of this evaluation is a measure of individual adaptation. The individuals with the best adaptation measure have more chances of reproducing and generating new individuals. Each individual (chromosome) is represented by a set of parameters (genes).

The GA uses two methods for generating new individuals: Crossover and Mutation. Once the next population has been generated, by means of Crossover, Mutation or both, it has to be evaluated, and it then replaces the earlier population. This process is repeated a finite number of times with the aim of obtaining the global optimum of the problem [8][9].

For optimizing multi-objective problems by the Pareto front approach, NSGA-II algorithm with some extensions to conventional GA is used. The NSGA-II main loop may be described as given in Fig. 1

1- Initialize population P_0
2- Sort(P_0)
3- While (not terminate)
4- Use selection, mutation and crossover to generate children Q_i
5- $P_i = P_i \cup Q_i$
6- Sort(P_i)
7- $P_i = P_i[1..popsize]$
8- $P_{i+1} = P_i$
9- i++
10- End While

Fig. 1 NSGA-II main loop

Steps 4 (selection) chooses and compares the individuals for the mating pool using tournament selection. Step 6 implicitly uses a comparison criterion to calculate the non-dominated fronts. In the following subsections a dominance relation for interval-valued multiple-objective functions is defined, which will be used to compare individuals in steps 4 and 6 [10].

A. Solution Representation

The representation scheme in GA determines how the problem is structured and also influences the genetic operators that are used. Which kind of solution representation is used depends on characteristics of the optimization problem.

When using GA for graph partitioning problem, many scholars used integer encoding [11][12], which provides a convenient and natural way to express the mapping from representation to solution domain. With integer encoding, the interpretation of the solution representation for graph partitioning is straightforward, so the integer encoding is also used in this paper. The solution to graph partitioning is encoded into a vector of integers, where the ith element of an individual is k if object i is assigned to partition k. For example, the objects n = {1,2,3,…,10} are assigned to three partitions k={1,2,3}, and one possible individual is [1 2 2 3 1 2 2 3 1 1]. Then the corresponding graph partitioning is {1, 5, 9, 10}, {2, 3, 6, 7} and {4, 8}.

B. Fitness Function

A fitness function evaluation is incorporated to assign a value to each individual. This value is a figure of merit which is calculated by using any domain knowledge that applies. In principle, this is the only point in the algorithm that domain knowledge is necessary. The individuals are chosen using the fitness value as a guide, where those with higher fitness values are chosen more often.

In case of multi-objective or Pareto optimization there is not any one value to describe the quality but several different qualities. The user is supposed to choose the proper quality combination that best fit to his/her purposes.

To facilitate the selection operation in GA, the global minimization problem is usually changed into a global maximization problem. Through transforming Eqs. (4), the proper fitness function for the aggregate inter-partition relationship among all partitions can be obtained

$$F=\begin{cases} U_{Coupling}-\sum_{k=1}^{K}\sum_{j=1}^{n}\sum_{i=1,i\neq j}^{n} x_{ik}\left(1-x_{jk}\right)r_{ij} & \\ & C<U_{Coupling}, \\ 0, & C\geq U_{Coupling}. \end{cases} \tag{8}$$

Where $U_{Coupling}$ should select an appropriate positive number to ensure the objective value of all good individuals are positive in the feasible solution space. On the other hand, $U_{Coupling}$ can also be utilized to adjust the selection pressure of GA. When $U_{Coupling}$ is increased, the relative fitness of good individuals is reduced, so the selection pressure is decreased, which can prevent the evolution process from premature convergence to get trapped into local minimums. But a too large $U_{Coupling}$ will slow down the evolution process, therefore will increase the computing time.

Consider P_1, P_2,… are non-dominated fronts and P_1 is the pareto front of the population. NSGA-II assigns a dummy and unique value to each individual using a fitness sharing function such that the worst fitness value assigned to the individuals of P_i is better than the best fitness value assigned to P_{i+1}' ones. This uniqueness value is related to the distance between each solution and its two closest neighbors. Distance is calculated from the associated fitness values of Eqs. (6) and (8). So the proper fitness function for the graph partitioning problem can be obtained

$$F_{total} = Rank + \frac{F^2 + W^2}{2}. \tag{9}$$

Open Science Index, Computer and Information Engineering Vol:3, No:5, 2009 publications.waset.org/11319/pdf

C. Population Initialization

The GA starts with randomly selected initial population. Individuals are created by generating an array of random numbers in the range 1 to partition count. Individual must represent a valid partitioning solution satisfying the constraints. If any of the constraints is violated, then the generated individual is regarded as invalid and discarded. In our problem, a valid partitioning solution is balanced. The process is repeated until two times more individuals of the population size are generated. Then the first half of individuals with smaller objective values is selected as the initial population.

Considering the high cost of generating elitist individuals, a database of best solutions of various runs has been used. The solutions of the database are injected into the first generation of the next runs to have some elitist solutions in the beginning.

The number of partitions can be estimated by dividing the total number of objects by the defined capacity of each partition. If the initialization process either runs too slowly or cannot generate valid individuals, then the number of partitions should be added by one and the initialization routine is restarted.

D. Genetic Operators

Selection operator chooses and compares individuals for the mating pool using the fitness value as a guide, where those with higher fitness values are chosen more often. Selecting individuals based on fitness value is a major factor in the strength of GAs as search algorithms. In graph partitioning using NSGA-II, the selection operator is tournament selection. Tournament selection closely mimics mating competition in nature is to randomly pick a small subset of individuals (two or three) from the mating pool, and the individual with the highest cost in this subset becomes a parent. The tournament repeats for every parent needed. In addition to the selection function, the best solution found so far during the search always survives to the next generation and replaces the worst solution. In this respect, all non-dominated solutions discovered by a multi-objective GA are considered elite solutions.

GA uses crossover to be performed on the parents selected in the selection process to create offspring. The purpose of crossover is to maintain the qualities of the solution set, while exploring a new region of the feasible solution space. For the graph partitioning problem, the double-point crossover operator is used: two crossover points are generated uniformly in the mated parents at random, and then the two parents exchange the middle portion between these crossover points to create two new offspring (Fig. 2).

Parent#1: 2 5 1|4 0 9|3 7 8 → Offspring#1: 2 5 1|7 6 8|3 7 8
Parent#2: 5 9 2|7 6 8|3 4 0 → Offspring#2: 5 9 2|4 0 9|3 4 0

Fig. 2 Double-point crossover operator

Each offspring must meet the same constraints as its parents. If any of constraints is violated, then both offspring are discarded and the crossover operation for the mated parents is retried. If the valid offspring cannot be obtained after 20 retries, the crossover operation for these two parents is given up to avoid a possible infinite loop.

The mutation operator occurs a short period of time after crossover and as in nature it exchanges two randomly selected positions of an individual. The purpose of mutation operation is to add diversity to the solution set and to avoid losing useful information in the evolution process. From another point of view, the mutation operation can improve the local search performance of GA. Together with crossover operation they complete the local and global search of the solution space.

For the graph partitioning problem, the stochastically selected object is randomly partitioned to a partition different from the original one (Fig. 3).

4 2 **6** 5 8 9 **0** 1 3 7 → 4 2 **0** 5 8 9 **6** 1 3 7

Fig. 3 Mutation operator

If any of constraints is violated, then the offspring is discarded and the mutation operation for the parent is retried. Because the mutation operation is carried out at a very small probability, the probability of generating invalid solution is also very small. Even if the invalid solution is generated, a valid solution can be easily obtained through a number of retries.

V. Simulation Results

The new evolutionary algorithm has been applied intensively to the multi-objective optimal graph partitioning of a real distribution system. 96 objects are needed to be partitioned into 4 partitions. The relationship matrix (see the Appendix) is generated by the gathered information from the mentioned system during 5 years. The relation between arbitrary two nodes is represented using floating point numbers.

Before optimizing the graph partitioning using the proposed GA method, a random method is firstly used to obtain a valid graph partitioning. The random method randomly generates 40 valid solutions, and then selects the solution with the minimum aggregate inter-partition relationship and maximum intra-partition relationship as the final solution. The aggregate inter-partition and aggregate intra-partition relationship can be calculated using Eqs. (4) and (6), respectively. Together with the graph partitioning they are written into Table 1.

This solution of NSGA-II method will be compared with the solution obtained by the proposed GA method, which can tell whether the proposed GA method really provides an improvement on the graph partitioning.

When the proposed GA method is used to optimize the graph partitioning, the GA parameters are set as follows: $Pop_{size} = 1000$, $P_c = 80$, $P_m = 2..6$, $Generation_{max} = 100$. We set $U_{Coupling}$ to the maximum inter-partition relationship value of all individuals in the initial population. Because the inter-partition relationship of individuals in the population will decrease with the evolution process, this method of determining $U_{Coupling}$ ensures the inter-partition relationship of all good individuals in later generations is positive.

TABLE I
THE GRAPH PARTITIONS BY RANDOM AND EXTENDED NSGA-II METHODS AND THEIR RESULTS

Partition method	Random method	Extended NSGA-II method
Graph partitions	{8,11,15,17,22,23,24,26,27,31,33,45, 47,55,60,64,67,72,86,90,92,93,95,96} {3,6,9,14,16,18,19,25,29,34,36,52, 58,59,65,66,68,71,78,79,80,81,83,94} {1,4,13,21,30,32,38,40,41,42,44,48, 50,51,53,56,63,74,76,77,82,84,85,91} {2,5,7,10,12,20,28,35,37,39,43,46, 49,54,57,61,62,69,70,73,75,87,88,89}	{6,13,21,24,25,26,27,33,34,36,38,39, 41,42,52,60,68,69,80,82,85,90,92,95} {3,8,10,15,17,20,23,37,40,44,47,51, 61,63,64,65,67,74,78,83,84,86,87,89} {4,7,9,11,12,14,16,18,19,22,28,29, 30,31,32,46,48,50,57,77,79,88,91,93} {1,2,5,35,43,45,49,53,54,55,56,58, 59,62,66,70,71,72,73,75,76,81,94,96}
Aggregate intra-partition relationship	33.14	53.78
Aggregate inter-partition relationship	119.15	98.50

TABLE II
THE GRAPH PARTITIONS OBTAINED IN FIVE OPTIMIZATION PROCESSES AND THEIR RESULTS

Run	P_c	P_m	Aggregate intra-partition relationship	Aggregate inter-partition relationship	Generation
1	80	2-3	42.43	109.86	17
2	80	3-6	40.90	111.39	15
3	80	2-3	53.78	98.50	13
4	80	3-6	50.40	101.89	16
5	80	2-3	53.11	99.18	14

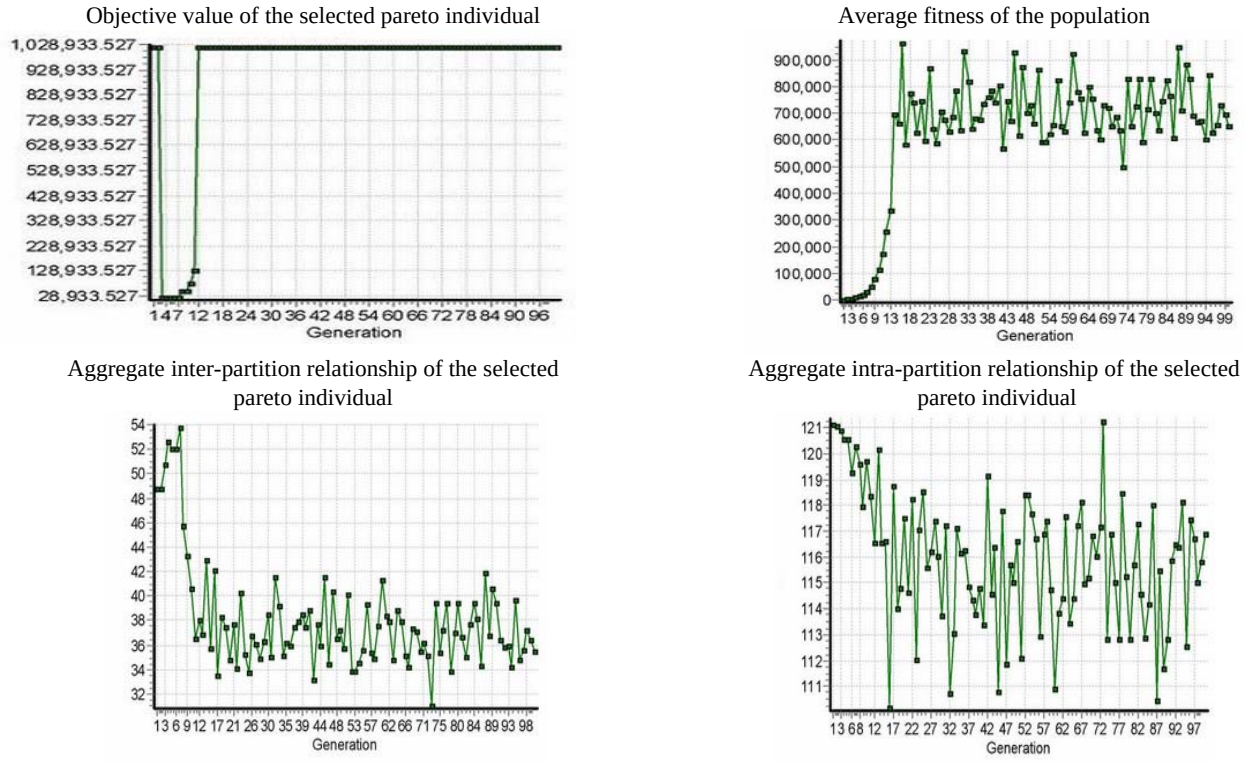


Fig. 4 Details of the evolution process corresponding to the selected Pareto solution

When selecting the final solution, we have to compromise between ‘‘aggregate inter-partition relationship’’ and ‘‘aggregate intra-partition relationship’’. We consider the aggregate inter-partition relationship is our main optimization objective, and simultaneously the aggregate intra-partition relationship needs to keep a large value. Corresponding to the selected Pareto solution, the graph partitioning, the aggregate inter-partition relationship, and the aggregate intra-partition relationship are also written into Table I. A simple contrast

Open Science Index, Computer and Information Engineering Vol:3, No:5, 2009 publications.waset.org/11319/pdf

shows that, the graph partitioning by the proposed GA decreases the aggregate inter-partition relationship that in the random graph partitioning.

Details of the GA evolution process corresponding to the final solution are depicted in Fig. 4. The objective of the selected best individual is monotonically increasing with the evolution process, and the average fitness of the population has a rapid increase at the beginning of the evolution process then fluctuations around a horizontal. The aggregate inter-partition relationship is generally decreasing and the aggregate intra-partition relationship is generally increasing with the evolution process, but also the converse at some generations, which reflect the two optimization objectives restraint and compete with each other in the evolution process. While one objective increases, the other objective decreases and vice versa.

For better solution, the whole optimization process is repeated for a number of times, and each time different crossover probability or mutation probability is set to the GA parameters. Graph partitions obtained in five repeated optimization processes and their relationship results are depicted in Table II, from which we selected the solution obtained in third optimization process, and the corresponding graph partitioning is depicted in Table I.

VI. Conclusion

In this paper the graph partitioning problem is firstly modeled, which is shown to be equivalent to a multi-objective optimization problem with constraints. Then a multi-objective GA method with some modifications to NSGA-II algorithm is proposed to optimize the graph partitioning. Finally the paper conducts a simulation research for the graph partitioning problem on a real distribution system. The simulation results confirm the effectiveness of the proposed multi-objective GA method. Multi-objective optimization of graph partitioning using hybrid cellular learning automata and genetic algorithm to combine the power of the GA with the speed of the local optimizer will be considered in our future research.

Appendix

In the appendix, a portion of the relationship matrix used in Section V is given.

	1	2	3	4	5	6	7	8
1	0	0.0289	0.1111	0.0676	0.6436	0.0541	0.531	0.7535
2	0.0289	0	0.1648	0.0793	0.0303	0.2196	0.0405	0.0243
3	0.1111	0.1648	0	0.1825	0.1378	0.1451	0.143	0.014
4	0.0676	0.0793	0.1825	0	0.0407	0.1687	0.1132	0.0828
5	0.6436	0.0303	0.1378	0.0407	0	0.0451	0.4394	0.683
6	0.0541	0.2196	0.1451	0.1687	0.0451	0	0.0278	0.014
7	0.531	0.0405	0.143	0.1132	0.4394	0.0278	0	0.6131
8	0.7535	0.0243	0.014	0.0828	0.683	0.014	0.6131	0
9	0.1517	0.1274	0.2338	0.1117	0.1467	0.1059	0.0184	0.1414
10	0.0151	0.1509	0.1303	0.1361	0.0144	0.1655	0.0207	0.0115
11	0.0158	0.1015	0.0429	0.0862	0.0246	0.1889	0.0036	0.0056
12	0.199	0.0194	0.0528	0.0312	0.2522	0.0109	0.3841	0.256
13	0.0065	0.0271	0.0537	0.0464	0.0066	0.0162	0.0285	0.0085
14	0.0225	0.056	0.1392	0.0851	0.0407	0.0486	0.0916	0.035
15	0.0016	0.0347	0.0093	0.02	0.0006	0.0251	0.0004	0.001
16	0.0016	0.0428	0.0151	0.0221	0.003	0.0407	0.0004	0.0035
17	0.0132	0.0009	0.0028	0.0005	0.0155	0.0007	0.0221	0.0145
18	0.024	0.0009	0.008	0.0009	0.0269	0.0003	0.0321	0.0215
19	0.0159	0.031	0.0299	0.039	0.0167	0.0246	0.0151	0.0111
20	0.0003	0.0075	0.0022	0.0103	2	0.0085	0.0017	0.0003
21	0.0011	0.0079	0.0083	0.0103	1	0.002	0.0017	0.0013
22	0.0019	0.0092	0.0108	0.0033	0.0018	0.0007	0.0026	0.0013
23	0.0003	0.0057	0.0009	0.007	2	0.0049	1	1
24	0.0008	0.0088	0.0114	0.008	2	0.0013	0.0004	0.0006
25	0.0008	0.0048	0.0012	0.0061	2	0.0033	1	0.0003
26	0.0008	0.0018	0.0003	0.0033	0.0009	0.0033	1	1
27	0.0013	0.0022	0.0012	0.0052	0.0012	0.0072	1	1
28	0.0027	0.0361	0.032	0.043	0.0042	0.0295	0.005	0.0018
29	0.0024	0.034	0.0307	0.0394	0.0039	0.0301	0.005	0.0018
30	0.0143	0.0004	0.0043	1	0.0164	1	0.0147	0.013
31	0.0057	0.007	0.0083	0.0023	0.0069	0.0003	0.0078	0.0054
32	0.0008	0.0088	0.0053	0.008	0.0003	0.0046	0.0004	0.0003
33	0.0019	0.0092	0.0043	0.0066	0.0003	0.0049	1	1
34	1	0.0057	0.0049	0.0103	2	0.0033	0.0004	0.0003
35	0.0008	0.0048	0.0012	0.0056	0.0006	0.0046	2	1
36	0.0008	0.0062	0.0006	0.0061	1	0.0042	0.0004	2
37	0.0008	0.0079	0.0034	0.0042	0.0003	0.0046	0.0009	0.0003
38	0.0011	0.0026	0.0006	0.0047	1	0.0036	1	1
39	0.0005	0.0018	0.0012	0.0052	0.0006	0.0046	1	1
40	0.0005	0.0031	0.0009	0.0047	0.0006	0.0059	1	1
41	0.0005	0.0026	0.0009	0.0047	0.0009	0.0069	2	2
42	0.0021	0.0142	0.0018	0.0127	0.0027	0.0338	1	1
43	0.1531	0.0096	0.0364	0.0102	0.1228	0.002	0.1754	0.1465
44	2	1	1	2	1	1	2	1
45	0.0016	0.0373	0.0067	0.0294	0.0021	0.041	1	2
46	0.0008	0.0268	0.0052	0.0277	0.0018	0.0267	2	1
47	0.018	0.0266	0.027	0.0202	0.0205	0.021	0.0231	0.0163
48	0.0029	0.0251	0.0027	0.0255	0.0021	0.0362	0.0004	0.0003
49	0.0005	0.0145	0.0012	0.0178	0.0018	0.0186	1	1
50	0.0008	0.0281	0.0015	0.0247	0.0009	0.0288	0.0008	1
51	0.0493	0.0022	0.0232	0.0033	0.0532	0.0007	0.0709	0.0374
52	0.0978	0.0113	0.0446	0.0084	0.1063	0.0051	0.1226	0.0801
53	0.0005	0.0259	0.0195	0.0176	0.0009	0.0105	0.0026	1
54	0.0011	0.0259	0.0185	0.0192	0.0012	0.0111	0.0026	0.0003

References

[1] Nguyen Thang, Ro Moon Byung, “Genetic Algorithm and Graph Partitioning”, IEEE Transaction on Computers 1996.

[2] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results.,"Evolutionary Computation, vol. 8, pp. 173–195, 2000.

[3] Coello Coello, Carlos A. A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques, Knowledge and Information Systems, Vol. 1, No. 3, pp. 269-308, August 1999.

[4] D.Goldberg. Genetic Algorithms in search, optimization and machine learning. Reading, Mass., Addison-Wesley, 1989.

[5] K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan, "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II," presented at Parallel Problem Solving from Nature VI Conference, Paris, France, 2000.

[6] Songerwala, M., 1994. Efficient solutions to the network division problem. Ph.D. Thesis, Department of Computer Science, Curtin University of Technology.

[7] Holland, J. H. 1975. Adaptation in Natural and Artificial Systems. Ann Arbor: University of Michigan Press.

[8] K.Bryant, Genetic Algorithms and the Traveling Salesman Problem, Thesis, Harvey Mudd College, Dept. of Mathematics, 2000.

[9] E. Cantu-Paz, A Survey of Parallel Genetic Algorithms, IlliGAL Report No. 97003, May 1997.

[10] Philipp Limbourg and Daniel E. Salazar Aponte. An Optimization Algorithm for Imprecise Multi-Objective Problem Functions, IEEE Congress on Evolutionary Computation, CEC2005, 2005.

[11] Krommenacker, N.; Rondeau, E.; Divoux, T.Factory Communication Systems, 2002. 4th IEEE International Workshop on Volume, Issue, 2002 Page(s): 149 – 156.

[12] Salcedo-Sanz, S.; Xin Yao Systems, Man, and Cybernetics, Part B, IEEE Transactions on Volume 34, Issue 6, Dec. 2004 Page(s):2343 – 2353.

Open Science Index, Computer and Information Engineering Vol:3, No:5, 2009 publications.waset.org/11319/pdf