T.C. DOĞUŞ ÜNİVERSİTESİ
INSTITUTE OF SCIENCE & TECHNOLOGY
Computer Information Sciences

Finding Best Performing Solution Algorithm for the QAP

Master Thesis

Burcu Müzeyyen Kıyıcığı
200791001

Advisor
Assoc. Prof. Ekrem Duman

June 2010
Istanbul

ACKNOWLEDGMENTS

I would like to thank my advisor Assoc. Prof. Ekrem Duman for his direction and guidance. Assoc. Prof. Ekrem Duman’s and Assist. Prof. Dr. Ali Fuat Alkaya’s recommendations and suggestions have been invaluable for my master thesis. Also I would like to thank TUBITAK for supporting my master thesis through the project 108M198.

Istanbul, June 2010 Burcu Müzeyyen KIYICIĞI

ABSTRACT

The quadratic assignment problem (QAP) of NP-Hard problems class is known as one of the hardest combinatorial optimization problems. In this thesis, a search is performed on the metaheuristics that have recently found widespread application in order to identify a heuristic procedure that performs well with the QAP. Algorithms which reflect implementations of Simulated Annealing, Genetic Algorithm, Scatter Search and Grasp – type metaheuristics are tested and using real test problems these algorithms are compared. Same set of algorithms are tested on general QAP problems and observation to identify successful algorithms is made. To conclude the best performing heuristic is not easy to name due to the fact that the performance of a heuristic depends on the context of the problem, which determines the structure and relationships of problem parameters.

ÖZET

Karesel atama problemi (KAP), NP-Zor sınıfına ait olup en zor kombinasyonel optimizasyon problemlerinden birisi olarak bilinir. Bu tez çalışmasında, KAP ile kullanılabilen en uygun sezgisel yöntemi tanımlayabilmek için yaygınca kullanılan meta sezgisel uygulamalar incelenmiş ayrıca Benzetimli tavlama, Genetik algoritma, Dağınık arama ve Açgözlü rassallaştırılmış uyarlamalı arama yordamı algoritmaları ile test edilmiş ve gerçek test problemleri ile karşılaştırılmıştır. Bunların dışında, aynı algoritmalar genel KAP problemlerinde test edilmiş ve hangi algoritmaların başarılı olduğu gözlemlenmiştir. Bu gözlemlere dayanarak özetlemek gerekirse, sezgisel algoritmaların performansı problemin içeriğine bağlı olduğu için ve bu da problemin yapısı ve parametreleri ile ilişkili olduğundan en iyi sezgisel algoritmayı tespit etmek oldukça güçtür.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

SYMBOLS

QAP - Quadratic Assignment Problem

KAP - Karesel Atama Problemi

SA - Simulated Annealing

GA - Genetic Algorithm

SS - Scatter Search

GX - Grasp

1. INTRODUCTION

The quadratic assignment problem (QAP), which is one of the hardest combinatorial optimization problems, is in the class of NP-Hard problems (Duman and Or, 2007). Many applications in several areas such as operational research, parallel and distributed computing, and combinatorial data analysis can be modelled by the QAP. Moreover, the QAP can be used to formulate other combinatorial optimization problems such as maximal clique, the travelling salesman problem, graph partitioning and isomorphism. The efficient heuristic methods known as Simulated Annealing, Genetic Algorithm, Scatter Search and Grasp can find a solution for the QAP. The goal of the thesis is to do a comparison of these methods from a QAP perspective.

1.1. The Quadratic Assignment Problem (QAP)

Matching n facilities with n locations causes a cost because of the interactions between the facilities. The main purpose of the quadratic assignment problem (QAP) is to minimize the total weighted cost. The following is a representation of the QAP.

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} c_{ij} d_{kl} x_{ik} x_{jl} \qquad (1.1.1)$$

$$\sum_{i=1}^{n} x_{ik} = 1, \qquad k = 1,\ldots,n$$

$$\sum_{k=1}^{n} x_{ik} = 1, \qquad i = 1,\ldots,n$$

$$x_{ik} \in \{0,1\} \qquad i,k = 1,\ldots,n$$

Equation 1.1.1: QAP formulation. (Koopmans and Beckmann, 1957)

Where x_{ik} = 1 means that k is assigned to i, d_{kl} is a matrix of flow of items to be transported between facilities and also c_{ij} is a matrix containing the distances or costs of transporting a single item between any two locations and also finding an assignment vector which minimizes the total transportation costs given by the sum of the product of the flow and distance between all pairs.

The QAP was first conceived as a mathematical model by Koopmans and Beckmann (1957) for economic activities, but it has been used in many different areas. It was applied in minimizing the total amount of connections between components in a backboard wiring by Steinberg (1961), in economic problems by Heffley (1972, 1980), in scheduling problems by Geoffrion and Graves (1976), in defining the best design for typewriter keyboards and control panels by Pollatschek et al (1976), in archeology by Krarup and Pruzan (1978), in parallel and distributed computing by Bokhari (1987), in statistical analysis by Hubert (1987), in the analysis of reaction chemistry by Forsberg et al (1994), in numerical analysis by Brusco and Stahl (2000). However, the most regarded application where QAP is used is the facilities layout problem. Dickey and Hopkins (1972) modelled the assignment of buildings in a University campus with QAP. Elshafei (1977) carried out a hospital planning and Bos (1993) used the QAP for similar reasons in planning of forest parks. A formulation of the facility layout design problem was done by Benjaafar (2002) to minimize work-in-process (WIP).

In spite of the fact that the QAP has been regarded as a popular model for facility layout problem, it is used in a wide range of area including chemistry, transportation, information retrieval, scheduling, statistical data analysis, parallel and distributed computing. As all the incoming and departing flights at an airport need to be directed to gates, assigning flights to gates can be modelled with QAP (Haghani and Chen, 1998). Allocation is an important logistic application of the QAP. Assigning rooms to persons by caring the undesirable neighbourhood constraints was formulated as a QAP by Ciriani, Pisanti, and Bernasconi (2004). To minimize the container rehandling operations at a shipyard, a generalization of the QAP was applied (Cordeau et al., 2005). Another location analysis problem which can be formulated as QAP is the parallel computing and networking (Gutjahr, Hitz, and Mueck, 1997; Siu and Chang, 2002). We consult Cela (1998) and Loiola et al (2007) to have in detailed knowledge about these and other applications of the QAP. Numerous other well-known combinatorial optimization problems such as the travelling salesman problem, the bin-packing problem, the maximum clique problem, the linear ordering problem and the graph-partitioning problem can be formulated as the QAP. It is easy to see that QAP has a variety of usage in many fields including transportation, manufacturing, logistics, economics, engineering, science, and sociology.

The QAP is known as a very complex problem since there is no algorithm to solve QAP instances with medium or large number of inputs. (Anstreicher et al., 2002) is one of the most successful approaches for the QAP implemented on a large grid which obtains optimal solutions for problems of size 30.

Many approaches have been implemented to QAP to overcome its solution difficulty.

1.2. Algorithms for the QAP

Seven of basic categories of approaches of QAP investigated in our research are Simulated annealing, Genetic algorithm, Scatter search, Grasp, Ant colony optimization, Tabu search and Variable neighbourhood search.

First algorithm we investigate is Simulated annealing which is an algorithm that feats the analogy which is made by solutions of combinatorial optimization problem to states of the physical system and cost related solutions to these states energies between optimization algorithms and statistical mechanics (Kirkpatrick et al., 1983).

Let E_i and E_{i+1} be energy states to two neighbour solutions.

$$\Delta E = E_{i+1} - E_i. \qquad (1.2.1)$$

Equation 1.2.1: Deterioration formula (Duman and Or, 2007).

These are the possible states which can occur: If ΔE is less than zero, continually energy reduction occurs which means problem cost function is reduced and the new allocation may be accepted. If ΔE is equal to zero, there is no change in the energy state and also problem cost function is not changed. If ΔE is more than zero, problem cost function is increased as well as the energy; to avoid poor local minima to come together the values which come from probability function is used.

One of the first applications of Simulated Annealing is proposed by Burkard and Rendl (1984) which is followed by Wilhelm and Ward (1987) adding new equilibrium components, other approach introduced by Abreu et al (1999), number of inversions of the problem solution and cost were reduced. Other approaches for simulated annealing applied to QAP are as follows: Bos (1993), Yip and Pao (1994), Burkard et al (1995), Peng et al (1996), Tian et al (1996, 1999), Mavridou and Pardalos (1997), Chiang and Chiang (1998), Misevicius (2000, 2003), Tsuchiya et al (2001), Siu and Chang (2002), Baykasoglu (2004), Duman and Or (2007).

Natural selection and adaptation are simulated by using Genetic Algorithms which generates a new solution by applying genetic operations on populations of initial solutions, cost is worked out and best solutions should be selected. More information about genetic algorithms can be found in: Brown et al (1989), Bui and Moon (1994), Tate and Smith (1995), Mavridou and Pardalos (1997), Kochhar et al (1998), Gong et al (1999). Other approaches for the genetic algorithms applied to QAP are Drezner and Marcoulides (2003), El-Baz (2004) and Wang and Okazaki (2005).

Glover (1977) introduced a Scatter Search method based on a study on linear programming problems. Therefore, method would take linear combinations of solution and generate new solution vectors in successive generations. Furthermore, metaheuristic is combination of initial phase where group of good solutions is referenced and evolutionary phase where new solutions are generated by using those references. Once the best generated solution is selected, it would be moved to reference set and then it will continue until a stop criterion is satisfied. Scatter search applications to the QAP can be found in Cung et al (1997). *The publications of Scatter search are much less than other algorithms (Loiola et al., 2007).*

Several researchers as follows applied the technique called (GRASP) which obtains approximate solution for the problem Li et al (1994), Feo and Resende (1995), Resende et al (1996), Fleurent and Glover (1999), Ahuja et al (2000), Pitsoulis et al (2001), Rangel et al (2000) and Oliveira et al (2004) built a GRASP using the path-relinking strategy, which looks for improvements along the paths joining pairs of good solutions.

Ant colony optimization (ACO) is a class of distributed algorithms which hold the definitions of properties of agents called ants. It is based on the idea how ants are able to make their way from colony to a food source. Ants work together in an ordinary activity to solve the problem. The main property of this method is these agents generate a synergetic effect since they work together and interact each other and the quality of solutions found increases. Numerical results for the QAP are presented in Maniezzo and Colorni (1995, 1999), Colorni et al (1996), Dorigo et al (1996) and Gambardella et al (1999) indicate that ant colony method generate few good solutions mainly close to each other therefore it is stated as a competitive metaheuristic.

Glover (1989) introduced a local search algorithm called Tabu Search to overcome integer programming problems with quality solutions where search process has list of best solutions with a priority value or an aspiration criterion, history of search process. Tabu list information and their priorities were used for new allocations in the neighbourhood to be accepted or rejected this way neighbourhood diversification and intensification was provided. Adaptations of this mechanism to QAP can be found in Skorin (1990, 1994), Taillard (1991), Bland and Dawson (1991), Rogger et al (1992), Chakrapani and Skorin-QAPov (1993), Misevicius (2003, 2005) and Drezner (2005). This mechanism was dependent on tabu list and how it is managed but it was shown that very good performance for QAP Taillard (1991) and Battiti and Tecchiolli (1994). Genetic algorithm and tabu search were compared by Taillard (1995) when they are applied to QAP.

Variable neighbourhood search (VNS) was proposed by Mladenovic and Hansen (1997). This search is systematic movement between set of neighbourhoods where there are a number of change rules defined that are used when there is no better solution in the current neighbourhood. VNS was applied to large combinatorial problems in Taillard and Gambardella (1999), three VNS strategies are introduced for the QAP.

There are other combined algorithms for the QAP. A combination of simulated annealing and genetic algorithm is presented by Bölte and Thonemann (1996). A combination of tabu search and simulated annealing is presented by Battiti and Tecchiolli (1994), Bland and

Dawson (1994), Chiang and Chiang (1998) and Misevicius (2001, 2004). Tabu search with a neural network was used by Talbi et al (1998) and Hasegawa et al (2002). And also tabu search and simulated annealing with fuzzy logic was used by Youssef et al (2003). A combination of genetic algorithm and tabu search with some hybrid algorithms is presented by Fleurent and Ferland (1994), Drezner (2003).

1.3. Methodology

In this research, a search is performed among those meta heuristic that have recently discovered widespread application which identifies a heuristic procedure that fits well with QAP. Specific algorithms reflecting implementations of Simulated Annealing, Genetic Algorithm, Scatter Search and Grasp – type meta heuristic were tested and compared using real data. The same set of algorithms is tested on common QAP problems as well. A sample Java application using Eclipse programming platform is made in order to compare the performance of meta-heuristics algorithms. There are eleven problems defined in (Duman and Or, 2007) eight of which we could obtain. This is the reason why only eight problems are tested. Deciding the best performing heuristic algorithm is complicated due to the fact that performance of a heuristic algorithm depends on the context of the problem which is determined by the structure and relationship of problem parameters.

1.4. Outline of Study

In the second part, we describe the formulation of the QAP and give information about algorithms of the QAP. In the third part, Simulated annealing, Genetic algorithm, Scatter search and Grasp are described with their parameters. In the fourth part, these algorithms' test results are presented. In the fifth part, find the best performing heuristic is tried to be identified structure and relationships of problem parameters.

2. ALGORITHMS IMPLEMENTED

This work is an experimental one. A sample Java application using Eclipse programming platform is made in order to compare the performance of meta-heuristics algorithms. These methods are Simulated annealing, Genetic algorithm, Scatter search and Grasp. (Duman and Or, 2007)'s literature problems are used for testing the methods. We present the best solution obtained by each method.

2.1. Simulated Annealing

In order to avoid local minima Simulated Annealing is used, it is one of the first algorithms which had a clear strategy that allowed worse quality solutions than the current solution to move, first an initial solution should be created randomly or heuristically, temperature parameter T should be initialized then N(s) is generated using solution s' at each value and current solution is accepted to be dependent on f(s), f(s') and T. Also s' replaces s if f(s') < f(s) or, in case f(s') ≥ f(s), with a probability which is a function of T and f(s')-f(s). Boltzmann distribution shows the probability as follows

$e^{-\frac{f(s')-f(s)}{T}}$ (Blum and Roli, 2003).

Having T in high values at the beginning and exchanges in pairs helps to avoid a local optimum to be trapped. T approaches zero and iterations are less than maximum iteration number (Noi) when pair wise exchanges keep happening. Simulated annealing implementation is shown as Figure-2.1.1.

```
Algorithm BT
0.  bs = cs.random()
1.  while(T>0&&totalR<Noi){
2.      for(j=0;j<R;j++){
3.          ns=cs.neighbour();
4.          delta=ns.getCost()-bs.getCost();
5.          if(ns.getCost()<cs.getCost()){cs=ns;}
6.          else if(Math.random()<Math.exp(-Math.abs(delta)/T)){cs=ns;}
7.           if(ns.getCost()<bs.getCost()){bs=ns;}
             }
8.      T/=a;
9.      R*=b;
10.     totalR+=R; }
```

Figure-2.1.1: Pseudocode SA

The variables in simulated annealing implementations are: initial temperature (T), temperature decrease ratio (a), number of iterations at each temperature setting (R), increase ratio in iteration number at each setting (b), maximum iteration number (Noi), current solution (cs), neighbour solution (ns) and best solution (bs). The acceptance probability is set to $e^{(-\Delta/T)}$, where Δ is the deterioration level. Neighbour solution is achieved by pair wise exchange in the current solution.

In Figure-2.1.1, until T is bigger than zero or total R is less than maximum iteration number (Noi), the process continues in line (1-10). Until ‘j’ is less than ‘R’, the process continues in line (2-7). In the first line (0), current solution (cs) is chosen randomly and ‘cs’ is assigned to ‘bs’. The random neighbour (ns) is chosen by current solution (cs) in line (3). ΔE_{ns} and ΔE_{bs} are two energy successive states, corresponding to two neighbour solutions and calculate ΔE to the difference between ΔE_{ns} and ΔE_{bs} in line (4). If the cost function of neighbour solution (ns) is less than the cost function of current solution (cs), ‘ns’ is assigned to ‘cs’ in line (5). Otherwise, If random(0, 1) is less than exp(ΔE /T), ‘ns’ is assigned to ‘cs’ in line (6). If the cost function of neighbour solution (ns) is less than the cost function of best solution (bs), ‘ns’ is assigned to ‘bs’ in line (7). T is decreased by T/a in line (8). R is increased by R*b in line (9). R is added to Total R in line (10).

2.1.1. Parameters Tuned

There are some parameters as follows:

i. Initial temperature (T) is set to either 100 (accepting 10 percent worse solutions has a probability of 0.05) or 1000 (accepting 10 percent worse solutions has a probability of 0.75) (Duman and Or, 2007).

ii. Temperature decrease ratio (a) is set to either 1.1 or 1.5 (Duman and Or, 2007).

iii. Number of iterations at each temperature setting (R) is set to either 5 or 25 (Duman and Or, 2007).

iv. Increase ratio in iteration number at each setting (b) is set to either 1.1 or 1.5 (Duman and Or, 2007).

v. Iteration number (Noi) is set to either N^3 or $N^3/3$ (Duman and Or, 2007).

Depending on different combinations of parameters 12 different simulated annealing heuristics are established (Table 2.1.1).

Table 2.1.1 Parameter settings of SA.

Procedure	Noi	T	R	a	b
H5	$N^3/3$	100	5	1.5	1.1
H6	N^3	100	5	1.5	1.1
H7	$N^3/3$	100	20	1.5	1.1
H8	N^3	100	20	1.5	1.1
H9	$N^3/3$	100	20	1.1	1.5
H10	N^3	100	20	1.1	1.5
H11	$N^3/3$	1000	5	1.5	1.1
H12	N^3	1000	5	1.5	1.1
H13	$N^3/3$	1000	20	1.5	1.1
H14	N^3	1000	20	1.5	1.1
H15	$N^3/3$	1000	20	1.1	1.5
H16	N^3	1000	20	1.1	1.5

Parameters used are the same as (Duman and Or, 2007). 12 heuristic algorithms were tested on eight test problems. The results are given in (3.1).

2.2. Genetic Algorithm

Genetic algorithms are based on natural evolution principles (Holland, 1990) and search number of solutions which are generated randomly or heuristically. To produce offspring (new individuals of the next generation). The population is developed by genetic operators like mutation which provides random modifications of the chromosome and diversity and crossover which creates new offspring by combining two parents. The parents (individuals from the current generation) are chosen depending on their survival, the offspring is become of the values which survived more. (Beasley et al., 1993).

The fitness function which shows the quality of an individual should determine the value of an invidual's performance in the current population. The fitness value which is dependent on the objective value is changed by the normalization. The equation is as follows:

$$\text{fitness}_i = (f_i - f_{min}) \setminus (f_{max} - f_{min}) \qquad (2.2.1)$$

Equation 2.2.1: Fitness function (Omatu and Yuan, 2000).

Where
fi : The objective value of individual *i*.
*f*min : The minimum objective value of current population.
*f*max : The maximum objective value of current population.

While selection chooses chromosomes in the population depending on their relative fitness, selection function decides for the parents of the next generation based on their fitness. The fitness which is used for the actual selection process is based on the objective value.

As we observed, two types of selection methods are used in genetic algorithms: Roulette wheel selection and Tournament selection.

Roulette wheel selection is simple selection method where offspring strings are allocated using a roulette wheel; members are selected based on their fitness. Although these members are chosen proportionally, it does not mean that the fittest member always goes to the next generation (Mahdavi et al., 2009).

Tournament selection is a method where *n* individuals are chosen randomly where the best fitness cost goes to next generation. The process stops when maximum number of generations is reached (Dikos et al.,1997).

It is stated that Roulette wheel selection is better than Tournament selection (Dikos et al., 1997).

Crossover genetic operator combines two or more parents where data and the genes are swapped between chromosomes, where programming of chromosomes changes this way from one generation to the next.

As we observed, six types of crossover operator PBX, PMX, OX, CX, ULX and MPX are as follows:

PBX (Position Based Crossover): A set positions are selected randomly from one parent and copying the genes on these positions to the child, a child is produced the genes selected before can be deleted. As a result sequence of cities obtains the cities the child needs. Chromosomes can be placed into an unfixed position of the child from left to right in the order to create an offspring (Misevicius et al., 2005).

PMX (Partially Matched Crossover): Two crossing sites of two chromosomes are chosen randomly where a matching selection is identified, this matching selection is used for the interchange to take place, and alleles are placed to their new positions in the offspring (Misevicius et al., 2005).

OX (Order Crossover): Two chromosomes and two crossing sites are selected randomly and also matching selection is used to position-by-position exchange to happen apart from genes, when genes are repeated they will not be placed in chromosomes where genes are assigned from left to right (Misevicius et al., 2005).

CX (Cycle Crossover): First gene in the first chromosomes is assigned to new chromosomes, (1 point), gene in the second chromosomes which is selected using opposite part of first gene of the first chromosomes is assigned in new chromosomes (2 point) the same method should be applied to all other genes (Mawdeslev et al., 2003).

ULX (Uniform Like Crossover): Two parents are chosen randomly where same two positions are assigned to this position in the new chromosome. The method applies to the rest of the genes (Mahdavi et al., 2009).

MPX (Multiple Parent Crossover): This time more than two parents are chosen randomly, gene will be assigned in the chromosomes only if the same position same gene value and gene’s value number is more. This method is used by the other positions as well (Misevicius et al., 2005).

MPX is considered as the best cross over if the new algorithm is combination of tabu search and genetic algorithms (Misevicius et al., 2005).
PMX is stated as suitable for QAP and TSP problems in (Chan et al., 2006). Also PMX is better when it uses Roulette wheel selection than when it uses Tournament selection (Dikos et al.,1997).

PBX, OX and ULX are stated as the worst for QAP problems (Mahdavi et al., 2009; Ravindra et al., 2000).

Mutation is a genetic operator which combines one parent to reproduce a new child where information and genes are swapped in the chromosomes which helps programming of a chromosome to change between generations. Three types of mutation operator are used by genetic algorithms are: Change by near, Swap and Insert.

Change by near: Randomly choosing number (B) and then B's position is assigned in [1,n] and change (B+1)'s position (Ramkumar et al., 2009).

$$B(2\ \underline{1}\ \underline{3}\ 4) = 2\ \underline{3}\ \underline{1}\ 4 \qquad (2.2.1)$$

Example - 2.2.1: Change by Near

According to Example 2.2.1, choosing number (1), and then change (B+1)'s position by means that change number (1) and (3).

Swap: Randomly choosing numbers (B1, B2) and then B1's position and B2's position is assigned in [1,n] and change (B1 and B2) (Ramkumar et al., 2009).

$$B(\underline{2}\ 1\ 3\ \underline{4}) = \underline{4}\ 1\ 3\ \underline{2} \qquad (2.2.2)$$

Example - 2.2.2: Swap

According to Example 2.2.2, choosing number (2, 4), these positions are assigned and change (2, 4).

Insert: Randomly choosing numbers (B1, B2), B1's position and then B2's position are assigned in [1,n] and then B2's value is inserted in B1's position and others change +1 position to right side (Tiwari et al., 2000; Ramkumar et al., 2009).

$$B(2\ \underline{1}\ 3\ \underline{4}) = 2\ \underline{4\ 1\ 3} \qquad (2.2.3)$$

Example - 2.2.3: Insertion

According to Example 2.2.3, choosing number (1, 4), these positions are assigned and insert (4) into 1's position, and other change + 1 position to right side.

Insert mutation is stated as better than mutation operators (Ramkumar et al., 2009; Ravindra et al., 2000). Change by near mutation method is considered as not suitable for QAP problems (Chan et al., 2006).

Table 2.2.1 GA- parameters used

Reference	Population Size	Generation Size	Probability of Crossover	Probability of Mutation	Runs
El-Baz, 2004	100	40	0.90	0.10	10
Mahdavi et al, 2009	50 – 200	Change	0.70 – 0.80	0.01 – 0.10	10
Gong et al, 1999	40	1200	0.40	0.20	10

In Table 2.2.1, the following observations can be made: the maximum size of population is 200, generation size is changeable and also the probability of crossover is between 0.70 and 0.80 and the probability of mutation is between 0.01 and 0.10.

Stop Rule: There are two points where the experiment can be stopped. One is when maximum number of generations is reached (El-Baz, 2004). The other is best value becomes stable (Mantawy et al., 1999).

Two genetic algorithm codes written with different selection methods being used, the selection methods used are SF and SS for producing new generation. SF is a fitness function where best number of solutions individual from all solutions is chosen where as SS function is a survival function where calculating the total change by taking minimum fitness value away from fitness of each individual and summing up all. This way worst solution has a chance. Genetic algorithm is shown in Figure-2.2.1.

```
Algorithm GA

1.  Population (pop) = create.random()
2.  bs ← pop.Min().getCost()
3.  While ( gs != G ||  bs != newbs)
     {
4.          newbs ← bs
5.          x  = pop.selectbyRouletteWheel()
6.          y  = pop.selectbyRouletteWheel()
7.          PMX (x,y) → c, n
8.          c   = MutatebyInsert(c)
            n   = MutatebyInsert(n)
9.          pop = pop + (c , n)
10.         10.a   newpop = SF (pop)       Or
            10.b.  newpop = SS (pop)
11.         pop.clear()
12.         pop = newpop
13.         bs ← pop.Min().getCost()
14.  }

10.a  SF  { for (int k=0; k < P ; k++) {
                 z = pop.Min().getFitness()  }
                 newpop.add(z)
          }

10.b  SS  { for (int k=0; k < P ; k++) {
                 z = pop.selectbyRouletteWheel() }
                 newpop.add(z)
          }
```

Figure-2.2.1: Pseudocode GA

The variables in genetic algorithm implementations are: population (pop), new population (newpop), individuals (x, y and z), children (c and n), old best solution (bs), new best solution (newbs), number of generation (gs), fitness function (SF), survival function (SS), maximum number of generation (G), number of solution (P), probability of crossover (CXP) and probability of mutation (MP).

In Figure 2.2.1, until number of solution (P) is reached, initial solutions are randomly created and these solutions are assigned to ‘pop’ in line (1). One solution had the lowest cost of population (pop) and this solution is assigned to old best solution (bs) in line (2). Until stopping rule (maximum number of generations is reached or best solution becomes stable) is met, the process continues in line (3-14). ‘bs’ is assigned to ‘newbs’ in line (4). A solution (x) is selected by Roulette Wheel method in line (5). A solution (y) is selected by Roulette Wheel method in line (6). Two children (c and n) are created by making PMX crossover with x and y in line (7). These solutions (c and n) are mutated by Insert in line (8) and ‘c’ and ‘n’ are added to population (pop) in line (9). Two genetic algorithm codes written with different selection methods being used, the selection methods used are SF (line 10.a) and SS (line 10.b) for producing new generation. In line (10.a), until number of solution (P) is reached, new generations are selected by using fitness function (SF). New generations are added to ‘newpop’. In line (10.b), until number of solution (P) is reached, new generations are selected by using survival function (SS). New generations are added to ‘newpop’. In line (11), ‘pop’ is deleted. ‘newpop’ is assigned to ‘pop’ in line (12). One solution had the lowest cost of population (pop) and this solution is assigned to old best solution (bs) in line (13).

2.2.1. Parameters Tuned

There are some parameters as follows:

i. Selection (S) is set to either (1. genetic code : SF - fitness function or
2. genetic code : SS - survival function).

ii. Mutation probability (MP) is set to 0.05.

iii. Crossover probability (CXP) is set to 0.75.

iv. Maximum Generation (G) is set to either 40 or 400.

v. Number of Solutions (P) is set to either 50,100 or 200.

Depending on different combinations of parameters 12 different genetic algorithm heuristics are established (Table 2.2.1.1).

Table 2.2.1.1 Parameter settings of GA.

Procedure	S	G	P
G1	SF	40	50
G2	SF	40	100
G3	SF	40	200
G4	SF	400	50
G5	SF	400	100
G6	SF	400	200
G7	SS	40	50
G8	SS	40	100
G9	SS	40	200
G10	SS	400	50
G11	SS	400	100
G12	SS	400	200

12 heuristic algorithms were tested on eight test problems. The results are given in (3.2).

2.3. Scatter Search Algorithm

Scatter search is an algorithm which does not allow same solutions and it is developed by Glover in 1997. Due to duplicate solutions not being allowed in scatter search, diversification should be considered very important. Applications of scatter search algorithms are investigated in Cung et al (1997).

Scatter search method starts with generating initial population. The fundamental idea is not to allow duplicate solutions in Reference set (RefSet). Because of this, diversification method is used in the population. Generation method is used to in the reference set where Refset has b1 (better solution of population) and b2 (high diverse solution of population). The reference set (RefSet) is developed by mutation operator. When there is no new solution in RefSet, the experiment should be stopped.

Scatter search consists of six methods: diversification method, improvement method, generation method, combine method/ pmx, reference set update method.

Generation method is used in the reference set where Refset has b1 (better solution) and b2 (high diverse solution), reference set can be updated when a new solution which satistifies the conditions as follows: new solution should have better cost function than the solution with worst function in b1, also should be high divert to the reference set than the solution with the worst divert in b2 which classifies b1 solutions as high quality and b2 solutions as high diversity (Marti et al., 2006).

High Diverse Solution: The more distance between two solutions, the solutions will have the high diversity. The distance is defined as shown in Equation 2.3.1.

$$d(p,q) = \Sigma^{n-1}{}_{i=1} \text{ distance between } p_{i+1} \text{ and } p_i \text{ in } q \qquad (2.3.1)$$

Equation 2.3.1: Distance equation (Yuan et.al, 2007).

According to Equation 2.3.1, calculating distance between p_{i+1} and p_i in q.

Diversification Method: As stated earlier diversification method is important since scatter search does not allow duplications in the reference set, this is the reason why tabu list is being used. The tabu list helps to reduce the computational effort to check duplication solutions by every solution to be checked while they are in the tabu list, if the solution is not a member of tabu list then it should be added to the list, if not the solution should be discarded (Cung et al., 1997 ; Marti et al., 2006).

Improvement Method: Two improvement methods called Swap and Insert are used in this research; Swap and Insert mutation operator are defined as (2.1).

Selection Method: Roulette wheel selection and lexicographical order selection methods are used in this research from selection methods. Roulette wheel selection is defined as in (2.1). Lexicographical order selection is a simple sort method; the name of the method indicates the strings are compared in alphabetical order / numerical order, from left to right (Marti et al., 2006).

Crossover Method: Two crossover methods are used PMX (Partially Matched Crossover) and Combine method. PMX (Partially Matched Crossover) is defined in (2.1). Combine Method is a crossover operator which produces new solutions from pairs of reference solutions where at least one new solution exists in the pair. The combine method has three steps starting with choosing a start node then move left part of the solutions “p1” and “p2” to the right end position. Then, first node of the parents is selected as combine solution “c”. Second step is to compare the distance last selected node in “c” to the next node of “p1” and “p2”, then short ones node should be chosen for the new rotation to “p1” and “p2”. Third step which is repeating step 2 happens until combine solution “c” is finished (Yuan, et al., 2007).

Reference Set Update Method is a method where Refset and new solutions are placed in the tabu list, then b1 (better solution) and b2 (high diverse solution) are selected, then Refset should be updated with b1 (better solution) and b2 (high diverse solution) (Marti et al., 2006).

Stop Rule: The experiment should be stopped when there is no new solution in RefSet (Marti et al., 2006).

Two scatter search algorithm codes provided in this research are (SPR and SLC), SPR algorithm code uses roulette wheel selection method, pmx crossover method and insert mutation method where as SLC consists of the following methods: lexicographical order selection method, combine crossover method and swap mutation method. Scatter search algorithm’s codes are shown in Figure-2.3.1 (SPR) and Figure-2.3.2 (SLC).

```
Algorithm SS – SPR

1.   Population (p) = create.random()
2.   Diversification.method (p)
3.   p =MutatebySwap(p)
4.   CheckedbyTabuList (p)
5.   RefSet = TabuList (b1) + TabuList (b2)
6.   While (New Solution isn't in the RefSet)
7.   {
8.           TabuList.clear()
9.            x= SelectedByRouletteWheel()
10.           y= SelectedByRouletteWheel()
11.           PMX (x,y) → z, t
12.           z = MutateByInsert (z)
13.           t =  MutateByInsert (t)
14.           TabuList = RefSet
15.      if (z.getCost() < t.getCost() && z ¢ TabuList)
16.      { TabuList = TabuList + z }
17.      if (z.getCost() > t.getCost() && t ¢ TabuList)
18.      { TabuList = TabuList + t }
19.       RefSet.update (b1 + b2)
20.   }
```

Figure-2.3.1: Pseudocode SS (SPR)

The variables in scatter search SPR implementations are: population (p), individual solutions (x and y), solutions (z and t), tabu list size (Tsize), number of better solutions (b1) and number of high diverse solutions (b2).

In Figure 2.3.1, there is no solution in the tabu list. Until Tsize is reached, initial solutions are randomly created and these solutions are assigned to ‘p’ in line (1). Each solution is checked by using diversification method that if a solution isn’t an element of tabu list, a solution adds to tabu list, otherwise discard this solution in line (2). Each solution is mutated by Swap in line (3). Each solution is checked by using diversification method that if a solution isn’t an element of tabu list, it adds to tabu list, otherwise discard this solution in line (4). Therefore, there are no duplicate solutions in tabu list. The reference set (RefSet) has b1 (better solutions of tabu list) and b2 (high diverse solutions of tabu list) in line (5). This process continues until there is no new solution in RefSet in line (6). Tabu List is cleared in line (8). The pairs (x, y) are selected by roulette wheel method in line

(9- 10) and solutions (z and t) are created by using PMX crossover with pairs (x and y) in line (11). These solutions (z, t) are mutated by Insert in line (12 – 13). RefSet adds to TabuList in line (14). If the fitness cost of z is less than the fitness cost of t and z isn't an element of Tabu List, z will add to Tabu List in line (15 - 16). If the fitness cost of z is higher than the fitness cost of t and t isn't an element of Tabu List, t will add to Tabu List in line (17 - 18). Tabu List consists of (z or t) and RefSet. Therefore, there is no duplicate solution in Tabu List. RefSet is updated by consisting of b1 (better solutions of tabu list) and b2 (high diverse solutions of tabu list) in line (19).

Algorithm SS – SLC

```
1.  Population (p) = create.random()
2.  Diversification.method (p)
3.  p =MutatebySwap(p)
4.  CheckedbyTabuList (p)
5.  RefSet = TabuList (b1) + TabuList (b2)
6.  While (New Solution isn't in the RefSet)
7.  {
8.       TabuList.clear()
9.        x= SelectedByLexicographicalorder()
10.       y= SelectedByLexicographicalorder()
11.       Combine (x, y) → z
12.       z = MutatBySwap (z)
13.       TabuList = RefSet
14.       if (z ¢ TabuList) {
15.            TabuList = TabuList + z }
16.        RefSet.update (b1 + b2)
17. }
```

Figure-2.3.2: Pseudocode SS (SLC)

The variables in scatter search SLC implementations are: population (p), individual solutions (x and y), solution (z), tabu list size (Tsize), number of better solutions (b1) and number of high diverse solutions (b2).

In Figure 2.3.2, there is no solution in the tabu list. Until Tsize is reached, initial solutions are randomly created and these solutions are assigned to 'p' in line (1). Each solution is checked by using diversification method that if a solution isn't an element of tabu list, a solution adds to tabu list, otherwise discard this solution in line (2). Each solution is mutated by Swap in line (3). Each solution is checked by using diversification method that if a solution isn't an element of tabu list, it adds to tabu list, otherwise discard this solution in line (4). Therefore, there are no duplicate solutions in tabu list. The reference set (RefSet) has b1 (better solutions of tabu list) and b2 (high diverse solutions of tabu list) in line (5). This process continues until there is no new solution in RefSet in line (6). Tabu List is cleared in line (8). The pairs (x, y) are selected by lexicographical order in line (9-10) and solution (z) is created by using Combine crossover with pairs (x and y) in line (11). This solution (z) is mutated by Insert in line (12). RefSet adds to TabuList in line (13). If z isn't an element of Tabu List, z will add to Tabu List in line (15). Tabu List consists of (z) and RefSet. Therefore, there is no duplicate solution in Tabu List. RefSet is updated by consisting of b1 (better solutions of tabu list) and b2 (high diverse solutions of tabu list) in line (16).

2.3.1. Scatter Search Algorithm – Parameters

There are some parameters as follows:

i. Tabu List Size (Tsize) is set to either 30, 50 or 100. The tabu list is used to reduce the computational effort of checking for duplicated solutions and also tabu list doesn't allow duplicate solutions. Therefore, Tsize is nearly bigger than sum of b1 and b2 (Dai, 2008).

ii. Number of high quality solutions (b1) is set to either 10 or 20. b1 has been taken more than 20 (Dai, 2008).

iii. Number of high diverse solutions (b2) is set to either 10 or 20. b2 has been taken more than 20 (Dai, 2008).

iv. Mutation probability (MP) is set to 0.05.

v. Crossover probability (CXP) is set to 0.75.

Depending on different combinations of parameters 14 different scatter search heuristics are established. (Table 2.3.1).

Table 2.3.1 Parameter settings of SS

Procedure	Type	Tsize	b1	b2
S1	SPR	30	10	10
S2	SPR	50	10	10
S3	SPR	50	20	10
S4	SPR	50	10	20
S5	SPR	50	20	20
S6	SPR	100	10	10
S7	SPR	100	20	20
S8	SPR	100	20	10
S9	SPR	100	10	20
S10	SCL	30	10	10
S11	SCL	50	10	10
S12	SCL	100	10	10
S13	SCL	50	20	20
S14	SCL	100	20	20

14 heuristic algorithms were tested on eight test problems. The results are given in (3.3).

2.4. Grasp Algorithm

Greedy randomized adaptive search procedure (GRASP) is a method where at each step an approximate solution is provided and best solution is generated from all solutions Grasp starts with generating an initial population and initial solutions are created randomly until the number of solutions is reached. The fundamental idea is to create restricted candidate list (RCL) because of selecting the partial solutions which are developed by mutation operators like Insert and Swap (Nehi and Gelareh, 2007). Grasp algorithm is shown in Figure-2.4.1.

Selection Method: roulette wheel selection method is used and defined in (2.1). Improvement Method: swap and insert mutate operators are used and defined in (2.1). *RCL* consists of partial solutions (Nehi and Gelareh, 2007).

Algorithm GX

```
1.   Create_initial_population(pop)
2.   oldv ← pop.getMin().getCost()
3.   for ( i  <  maxi )
4.   {
5.             Create_RCL (
6.                       for(int i=0;i<population.size()/2;i++)
7.                       {s1 = selectByRouletteWheel();
8.                            RCL.add(s1);
9.                            pop.remove(s1); }
10.                       )
11.            Update_Solution (
12.                       for(int k=0;k<RCL.size();k++)  {uprs.add()}
13.                          upra = uprs;
14.                          uprm = uprs;
15.                          uprs.mutateBySwap(mp);
16.                          uprm.mutateByInsertion(mp);
17.                       if(uprs.getCost() < uprm.getCost() ) {gbest ← uprs
18.                                                      else    gbest  ←uprm }
19.                       if (gbest < upra.getCost() )   { gbest ← gbest
20.                                               else  gbest  ← upra, pop.add(upra) }
21.                       if (gbest == uprs.getCost() ) { pop.add (uprs) }
22.                       if (gbest == uprm.getCost()) { pop.add (uprm) }
23.                       return bestv ← pop.Min().getCost()
24.                       )
25.       if (oldv   <  bestv)  {bestv ← oldv, else bestv ← bestv }
26.  }
```

Figure-2.4.1: Pseudocode GX

The variables in grasp implementations are: population (pop), solutions (s1, uprs, upra and uprm), restricted candidate list (RCL), iteration (i), maximum iteration (maxi), old best solution (oldv), best solution (bestv), temp best solution (gbestv) and number of solution (seed).

In Figure 2.4.1, initial solutions are created randomly and these solutions add to population (pop) in line (1). One solution has the lowest cost of population (pop) and this solution is assigned to ‘oldv’ in line (2). Until the maximum iteration (maxi) is reached, the process continues in line (3 - 26). *RCL list* is created in line (5- 10). Until population size is equal to half (population/2), these solutions are selected by roulette wheel method in line (7) and these solutions add to *RCL list* in line (8) and these solutions delete in the population (pop) in line (9). Solutions are updated in line (11 – 24). Until *RCL list* size is reached, the solutions of RCL list add to ‘uprs’ in line (12). The solution (uprs) is assigned to ‘upra’ in line (13). The solution (uprs) is assigned to ‘uprm’ in line (14). The solution (uprs) is mutated by Swap in line (15). The solution (uprm) is mutated by Insert in line (16). If (uprs)’s cost is less than (uprm)’s cost, ‘uprs’ is assigned to ‘gbest’ , otherwise ‘uprm’ is assigned to ‘gbest’ in step (17 – 18). If ‘gbest’ is less than (upra)’s cost, return gbest, otherwise ‘upra’ is assigned to ‘gbest’ and ‘upra’ adds to population (pop) in line (19 – 20). If ‘gbest’ is equal to uprs’s cost, ‘uprs’ adds to population (pop) in line (21). If ‘gbest’ is equal to uprm’s cost, ‘uprm’ adds to population (pop) in line (22). One solution which has the lowest cost of the population (pop), is set to ‘bestv’ in line (23). If ‘oldv’ is less than ‘bestv, return oldv else return bestv in line (25). When maximum iteration is reached, the experiment should be stopped.

2.4.1. Grasp Algorithm – Parameters

There are some parameters as follows:

i. Maximum iteration (MI) is set to 100, 1000, 2000 and 5000.

ii. Number of solutions (seed) is set to 50, 500, 1000 and 2500.

iii. Mutation probability (MP) is set to 0.05.

Depending on different combinations of parameters 10 different grasp heuristics are established. (Table 2.4.1).

Table 2.4.1 Parameter settings of GX

Procedure	MI	Seed
R1	100	50
R2	1000	500
R3	5000	500
R4	5000	2500
R5	5000	1000
R6	2000	500
R7	2000	1000
R8	2000	50
R9	2000	2500
R10	5000	50

10 heuristic algorithms were tested on eight test problems. The results are given in (3.4).

3. EXPERIMENTAL RESULTS

There are eleven problems defined in (Duman and Or, 2007)'s literature eight of which we have obtained. Due to this, only eight problems are tested in Laboratory of Dogus University.

Characteristics of these tables are shown in Table 3.1.

Table 3.1: The number of components in each problem.

Problems	#
PS11AK08-9	146
PS11AK1011	159
PS11AK12-7	152
PS11AK15-4	170
PS11AK16-3	261
PS11AK16-4	262
PS11AK16-5	280
PS11AK17N3	185

Table 3.1 shows number of components in departments.

Table 3.2: The context of test problem (PS11AK08-9).

PS11AK08-9 - Not Defteri

Dosya Düzen Biçim Görünüm Yardım

1	1635	21386	38	1
2	2298	21310	26	1
3	2628	21310	26	1
4	3568	21513	5	1
5	4248	21805	36	2

According to Table 3.2, PS11AK08-9 test has 5 different columns. First column contains row number and increases by 1. Second is x-coordinate, third is y-coordinate, fourth is the cell (department) where it is put and last is the type. It is the same for the rest of the test problems.

Detecting the best performing heuristic algorithm is complicated due to the fact that the performance of a heuristic is dependent on the problem type which is determined by the structure and relationship of problem parameters.

3.1. Simulated Annealing – Test Results

12 heuristic procedures on eight test problems have been run ten times in order to get detailed results. (See [Table 6.1] for a full tabulation in see Appendix). Out of eight test problems, the best average and the best solutions are displayed in (Table 3.1.1). Each test problem is run for 10 times and there are total 120 (12 x 10) results for each test problem. From 120 results the best solutions are indicated. (Table 3.1.1- Best solution) from each procedure for each test case (10 times) the average solutions are calculated (Table 3.1.1 - Average solution).

H8, H9, H15 and H16's results are worse than rest of the results. Therefore, the performance of (Noi = N^3, R=20 and T=1000) parameters are better than the performance of (Noi = N^3/ 3, R=5 and T=100) parameters.

Table 3.1.1 SA - best average and best solution of problems

Problem	Best Average (result, procedure)	Best Solution (result, procedure)
PS11AK08-9	799,00 H5	740,00 H14
PS11AK1011	806,60 H14	750,00 H13
PS11AK12-7	818,60 H13	774,00 H13
PS11AK15-4	746,80 H7	714,00 H6,H11
PS11AK16-3	1395,60 H14	1348,00 H12
PS11AK16-4	1441,60 H10	1392,00 H6,H14
PS11AK16-5	1502,20 H10	1454,00 H6
PS11AK17N3	1156,20 H7	1076,00 H14

Parameters of each procedure and successful results for testing are proposed in (Table 3.1.2). This table is supported by (Table 3.1.1). For instance out of eight problems there is only one best average and none of the best solutions in H5 is in (Table 3.1.1). Looking at H5, best average is 1 and best solution is 0 (Table 3.1.2). This value is used to compare the performance of procedures.

We have come up with the idea that the highest number of "best solution" is the best heuristic procedure. This is the reason why time is not a concern on this research, each procedure only works 10 times, and if run time is critical, the highest number of "best average" is the best heuristic procedure.

Table 3.1.2 SA - test results and parameters used

Procedure	Best Average Number	Best Solution Number	Noi	T	R	A	B
H5	*1*	0	$N^3/3$	100	5	1.5	1.1
H6	0	3	N^3	100	5	1.5	1.1
H7	*2*	0	$N^3/3$	100	20	1.5	1.1
H8	*0*	0	N^3	100	20	1.5	1.1
H9	*0*	0	$N^3/3$	100	20	1.1	1.5
H10	*2*	0	N^3	100	20	1.1	1.5
H11	*0*	1	$N^3/3$	1000	5	1.5	1.1
H12	*0*	1	N^3	1000	5	1.5	1.1
H13	*1*	2	$N^3/3$	1000	20	1.5	1.1
H14	*2*	3	N^3	1000	20	1.5	1.1
H15	*0*	0	$N^3/3$	1000	20	1.1	1.5
H16	*0*	0	N^3	1000	20	1.1	1.5

Looking at Table 3.1.2, H14 has 2 of best average and 3 of best solutions, whereas H14 is the best procedure, as per the results the performance of (Noi = N^3, R=20, a=1.5, b=1.1

and T=1000) parameters are the best. It is observed that using higher initial temperature and higher number of iterations have positive effect on the performance of the algorithm. A numerical comparison shows that N^3 iterations are better than $N^3/3$ with respect to heuristic procedures, which shows one percent benefit is observed more with H14 being best heuristic procedure in Duman and Or (2007).

3.2. Genetic Algorithm – Test Results

12 heuristic procedure on eight test problems have been run ten times in order to get the results. (See [Table 6.2] for a full tabulation in see Appendix). Out of eight test problems the best average and the best solutions are listed in (Table 3.2.1). Each test problem is run for 10 times and there are total 120 (12 x 10) results for each test problem. From 120 results the best solutions are given in (Table 3.2.1- Best solution) from each procedure for each test case (10 times) the average solutions are calculated (Table 3.2.1 - Average solution).

Between G1 and G12, G5 and G6 shows the best performance with parameters of (S = SF, G=400, P = 200) and also using SS (survival selection) G12 shows the best performance between G7 and G12.

Table 3.2.1 GA- best average and best solution of problems (G1-G12)

Problem	Best Average (result, procedure)	Best Solution (result, procedure)
PS11AK08-9	856,20 G6	806,00 G6
PS11AK1011	849,20 G6	802,00 G5
PS11AK12-7	917,20 G6	854,00 G5
PS11AK15-4	762,80 G6	716,00 G6
PS11AK16-3	1407,60 G6	1378,00 G6
PS11AK16-4	1490,60 G6	1458,00 G5
PS11AK16-5	1593,80 G5	1514,00 G5
PS11AK17N3	1230,40 G6	1154,00 G6

Parameters of each procedure and successful results for testing are proposed in (Table 3.2.2). This table is supported by (Table 3.2.1). For instance out of eight problems there is only one best average and four of the best solutions in G5 is in (Table 3.2.1). Looking at G5, best average is 1 and best solution is 4 (Table 3.2.1). This value is used to compare the performance of procedures.

Table 3.2.2 GA- test results and parameters used

Procedure	Best Average Number	Best Solution Number	S	G	P
G1	*0*	0	SF	40	50
G2	0	0	SF	40	100
G3	*0*	0	SF	40	200
G4	*0*	0	SF	400	50
G5	*1*	4	SF	400	100
G6	*7*	4	SF	400	200
G7	*0*	0	SS	40	50
G8	*0*	0	SS	40	100
G9	*0*	0	SS	40	200
G10	*0*	0	SS	400	50
G11	*0*	0	SS	400	100
G12	*0*	0	SS	400	200

G6 is the best heuristic procedure with 7 of best average and 4 of best solutions between G1 and G12. Per the results the performance of (S = SF, G = 400 and P = 200) parameters are best. It is observed that selection method (SF), higher initial population and higher number of generations have positive effect on the performance of the algorithm. New generations are created by fitness selection method SF where best number of solutions among all solutions is chosen without taking worst number of solutions into account. Otherwise, parameters of G12 (S= SS, G = 400 and P = 200) which has a better performance between G7 and G12. It is observed using higher initial population and higher number of generations has positive affect on the performance of the algorithm. New generations are created using SS (Survival selection) which allows worst solutions to be chosen. As a result fitness selection is better than survival selection.

3.3. Scatter Search – Test Results

14 heuristic procedure on eight test problems have been run ten times in order to get the results. (See [Table 6.3] for a full tabulation in see Appendix). Out of eight test problems the best average and the best solutions are listed in (Table 3.3.1). Each test problem is run for 10 times and there are total 140 (14 x 10) results for each test problem. From 140 results the best solutions are given in (Table 3.3.1- Best solution) from each procedure for each test case (10 times) the average solutions are calculated (Table 3.3.1 - Average solution).

S3, S5 and S7 show the best performance between S1 and S14. SPR shows better performance than SLC.

Table 3.3.1 SS - best average and best solution of problems (S1 - S14)

Problem	Best Average (result, procedure)	Best Solution (result, procedure)
PS11AK08-9	993,2 S5	870,0 S5
PS11AK1011	976,6 S5	858,0 S5
PS11AK12-7	1028,8 S5	886,0 S5
PS11AK15-4	835,8 S5	772,0 S5
PS11AK16-3	1614,6 S5	1284,0 S3
PS11AK16-4	1599,2 S7	1450,0 S7
PS11AK16-5	1802,6 S5	1540,0 S5
PS11AK17N3	1494,0 S5	1248,0 S5

Parameters of each procedure and successful results for testing are proposed in (Table 3.3.2). This table is supported by (Table 3.3.1).

Table 3.3.2 SS- test results and parameters used

Procedure	Best Average Number	Best Solution Number	Tsize	B1	B2
S1	0	0	30	10	10
S2	0	0	50	10	10
S3	0	1	50	20	10
S4	0	0	50	10	20
S5	7	6	50	20	20
S6	0	0	100	10	10
S7	1	1	100	20	20
S8	0	0	100	20	10
S9	0	0	100	10	20
S10	0	0	30	10	10
S11	0	0	50	10	10
S12	0	0	100	10	10
S13	0	0	50	20	20
S14	0	0	100	20	20

S5 is the best heuristic procedure with 7 of best average and 6 of best solutions. As per the results the performance of (Tsize = 50, b1 = 20 and b2 = 20) parameters are the best.

b2 will have an opportunity to choose worst solution when Tsize has a very high value (Tsize is bigger than sum of b1 and b2). The performance of SPR (S1 – S9) which uses roulette wheel selection method, pmx crossover method and insert mutation method, is better than SCL (S10 – S14) using leoxical order selection method, combine crossover method and swap mutation. According to SPR, roulette wheel selection method allows for worst solution (b) to perform PMX crossover therefore two solutions (a, c) are occurred and two solutions (a, c) are mutated by insert and solution which has the lowest cost of the three solutions (a, b or c) is added to population. Furthermore, number of worst solutions is less comparing to iterations. According to SCL, leoxicalorder selection chooses a solution orderly. One solution is produced by combined crossover of solutions. A solution is mutated by swap, and then a solution adds to population, where number of best solutions will be less than number of best solutions before iterations.

3.4. Grasp – Test Results

10 heuristic procedure on eight test problems have been run ten times in order to get the results. (See [Table 6.4] for a full tabulation in see Appendix). Out of eight test problems the best average and the best solutions are listed in (Table 3.4.1). Each test problem is run for 10 times and there are total 100 (10 x 10) results for each test problem. From 100 results the best solutions are given in (Table 3.4.1- Best solution) from each procedure for each test case (10 times) the average solutions are calculated (Table 3.4.1 - Average solution).

Table 3.4.1 GX - best average and best solution of problems (R1-R10)

Problem	Best Average (result, procedure)	Best Solution (result, procedure)
PS11AK08-9	939,0 R10	852 R10
PS11AK1011	936,6 R3	836 R10
PS11AK12-7	982,2 R4	884 R3
PS11AK15-4	804,8 R5	742 R4
PS11AK16-3	1527,8 R4	1444 R3
PS11AK16-4	1571,8 R5	1486 R5
PS11AK16-5	1687,2 R4	1556 R5
PS11AK17N3	1398,4 R3	1314 R3

Parameters of each procedure and successful results for testing are proposed in (Table 3.4.2). This table is supported by (Table 3.4.1).

Table 3.4.2 GX- test results and parameters used

Procedure	Best Average Number	Best Solution Number	MI	Seed
R1	0	0	100	50
R2	0	0	1000	500
R3	2	3	5000	500
R4	3	1	5000	2500
R5	2	2	5000	1000
R6	*0*	0	2000	500
R7	*0*	0	2000	1000
R8	*0*	0	2000	50
R9	*0*	0	2000	2500
R10	*1*	2	5000	50

R4 is the best heuristic procedure with 3 of average and 1 of best solution. The results show that the performance of (MI = 5000, Seed = 2500) parameters are the best.Using higher maximum iteration and higher number of solutions help performance of algorithm to increase.

4. COMPARISON OF ALGORITHMS

CM (Comparison of Methods) developed using Java is used to compare performance of Simulated Annealing, Genetic Algorithm, Scatter Search and Grasp algorithms. (Duman and Or, 2007)'s literature problems are tested using CM with each metaheuristic algorithms.

There are two comparisons side to tests, one is best procedure of methods and the other is comparing best solution and best average of each problem. In Part 3 comparisons of these algorithms are provided.

Comparison of Method's Best Procedure:

H14 is the best procedure of Simulated annealing. G6 is the best procedure of Genetic algorithm. S5 is the best procedure of Scatter search. R4 is the best procedure of Grasp.

Table 4.1.1 Best average of algorithms

Problem	Best Average				Best Average
	H14	G6	S5	R4	
PS11AK08-9	820	856,2	993,2	967,4	H14
PS11AK1011	806,6	849,2	976,6	939,4	H14
PS11AK12-7	842,2	917,2	1028,8	982,8	H14
PS11AK15-4	752	762,8	835,8	806,6	H14
PS11AK16-3	1395,6	1407,6	1614,6	1527,8	H14
PS11AK16-4	1461,2	1490,6	1684,2	1582,2	H14
PS11AK16-5	1542,6	1603,8	1802,2	1687,2	H14
PS11AK17N3	1165,4	1230,4	1494,0	1402,8	H14

The following observations are made from Table 4.1.1, H14 is the best average of all these problems. Simulated Annealing has best average for QAP test problems. According to that results, the performance of (Noi = N^3, R=20, a=1.5, b=1.1 and T=1000) parameters are best. Orderly, the performances of others are G6, R4 and S5.

Table 4.1.2 Best solution of algorithms

Problem	Best Solution				Best Solution
	H14	G6	S5	R4	
PS11AK08-9	740	806	870	904	H14
PS11AK1011	756	808	858	872	H14
PS11AK12-7	800	858	886	928	H14
PS11AK15-4	718	716	772	742	G6
PS11AK16-3	1370	1378	1398	1450	H14
PS11AK16-4	1392	1458	1502	1510	H14
PS11AK16-5	1474	1546	1540	1632	H14
PS11AK17N3	1076	1154	1248	1320	H14

Looking at Table 4.1.2, H14 is the best solution of 7 problems, but G6 is the best solutions of the PS11AK15-4 problems. Simulated Annealing has best solution for solving QAP test problems. Orderly, the performances of others are G6, S5 and R4.

Comparison of Each Problem's Best Solution and Best Average:

H5, H7, H10, H13 and H14 are the best average of each problem in Simulated Annealing. G6 and G5 are the best average of each problem in Genetic Algorithm. S5 and S7 are the best average of each problem in Scatter Search. R3, R4, R5 and R10 are the best average of each problem in Grasp.

Table 4.2.1: Best average of each problem in algorithms

Problem	Best Average				Best Average
	SA	GA	SS	GX	
PS11AK08-9	799 H5	856,2 G6	993,2 S5	939,0 R10	H5
PS11AK1011	806,6 H14	849,2 G6	976,6 S5	936,6 R3	H14
PS11AK12-7	818,6 H13	917,2 G6	1028,8 S5	982,8 R4	H13
PS11AK15-4	746,8 H7	762,8 G6	835,8 S5	804,8 R5	H7
PS11AK16-3	1395,6 H14	1407,6 G6	1614,6 S5	1527,8 R4	H14
PS11AK16-4	1441,6 H10	1490,6 G6	1599,2 S7	1571,8 R5	H10
PS11AK16-5	1502,2 H10	1593,8 G5	1802,6 S5	1687,2 R4	H10
PS11AK17N3	1156,2 H7	1230,4 G6	1494 S5	1398,4 R3	H7

Table 4.2.1 results are as: H5, H7, H10, H13 and H14 are the best average in each problem. Simulated Annealing has best average of all eight problems. Orderly, the performances of others are Genetic algorithm, Grasp and Scatter search.

H6, H11, H12, H13 and H14 are the best solutions of each problem in Simulated Annealing. G6 and G5 are the best solutions of each problem in Genetic Algorithm. S3, S5 and S7 are the best solutions of each problem in Scatter Search. R3, R4, R5 and R10 are the best solutions of each problem in Grasp.

Table 4.2.2: Best solution of each problem in SA, GA, SS, GX

Problem	Best Solution				Best Solution
	SA	GA	SS	GX	
PS11AK08-9	740 H14	806 G6	870 S5	852 R10	H14
PS11AK1011	750 H13	802 G5	858 S5	836 R10	H13
PS11AK12-7	774 H13	854 G5	886 S5	884 R3	H13
PS11AK15-4	714 H6, H11	716 G6	772 S5	742 R4	H6,H11
PS11AK16-3	1348 H12	1378 G6	1284 S3	1444 R3	S3
PS11AK16-4	1392 H6,H14	1458 G5	1450 S7	1486 R5	H6,H14
PS11AK16-5	1454 H6	1514 G5	1540 S5	1556 R5	H6
PS11AK17N3	1076 H14	1154 G6	1248 S5	1314 R3	H14

According to Table 4.2.2, the best solutions are H6, H11, H13, H14 and S3. Simulated Annealing has best solutions of 7 problems, but a PS11AK16-3 problem is S3 as Scatter search.

4.1. Summary

In this study a program called CM is developed to compare performance of metaheuristic algorithms in Eclipse platform. Algorithms compared are simulated annealing, genetic algorithm, scatter search and grasp. (Duman and Or, 2007)’s literature problems are solved using these algorithms and compared by CM. As stated before, deciding the best performing heuristic is complicated by the fact that the performance of a heuristic depends on the context of the problem, which is determined the structure and relationships of problem parameters.

Total of 48 heuristic procedures including 12 SA, 12 GA, 14 SS and 10 GX are all investigated. These heuristic algorithms are tested on 8 real data. The best performing algorithm, H14, is chosen to be used in the iterative solution methodology of QAP. As per the results the performance of (Noi = N^3, R=20, a=1.5, b=1.1 and T=1000) parameters are best. H14 is also found as best heuristic procedure in Duman and Or (2007).

Comparing best procedure of methods (SA- H14, GA- G6, SS – S5, GX – R4), H14 is the best average of all 8 problems and the best solution of 7 problems. Simulated Annealing has best procedure to solve QAP test problems.

Comparing best solutions of each problem, Simulated Annealing is the best metaheuristic algorithm for solving QAP problems, Duman and Or (2007) has the same result in finding the best algorithm for QAP problems. Orderly, the performances of other algorithms are Genetic algorithm, Grasp and Scatter search.

5. REFERENCES

[1] Ahuja, R.K., Orlin, J.B. and Tiwari, A., (2000), "A greedy genetic algorithm for the quadratic assignment problem", Computers & Operations Research 27, 917-934.

[2] Chan, F.T.S., Lau, K.W., Chan, P.L.Y. and Choy, K.L., (2006), "Two-stage approach for machine-part grouping and cell layout problems", Robotics and Computer-Integrated Manufacturing 22, 217–238.

[3] Chwif, L., Barretto, M.R.P. and Moscato, L.A., (1998), "A solution to the facility layout problem using simulated annealing", Computers in Industry, 36,125-132.

[4] Cung, V.-D., Mautor, T., Michelon, P., Tavares, A., 1997. A Scatter Search Based Approach for the Quadratic Assignment Problem. In: Bäck, T., Michalewicz, Z., Yao, X. (Eds.), Proceedings of IEEE-ICEC-EPS'97, IEEE International Conference on Evolutionary Computation and Evolutionary Programming Conference, pp. 165–170.

[5] Dikos, A., Nelson, P.C., Tirpak, T.M. and Wang, W., (1997), "Optimization of high-mix printed circuit card assembly using genetic algorithms," Annals of Operations Research 75,303 – 324.

[6] Drezner, Z., (2008), "Extensive experiments with hybrid genetic algorithms for the solution of the quadratic assignment problem", Computers & Operations Research, 35, 717-736.

[7] Duman, E. and Or, I ., (2007), "The quadratic assignment problem in the context of the printed circuit board assembly process", Computers & Operations Research 34,163-179.

[8] El-Baz, M.A., (2004), "A genetic algorithm for facility layout problems of different manufacturing environments",Computer & Industrial Engineering 47, 233-246.

[9] Gong, D., Yamazaki, G., Gen, M., and Xu, W., (1999), "A genetic algorithm method for one-dimensional machine location problems", Int. J. Production Economics 60-61, 337-342.

[10] Koopmans, T. and Beckmann, M. ,(1957), "Assignment problems and the location of invisible economic activities", Econometrica ,53 – 76.

[11] Lim, M.H. and Omatu, S., (2000), "Efficient genetic algorithms using simple genes exchange local search policy for the quadratic assignment problem", Computational Optimization and Applications, 15, 249-268.

[12] Loiola, E.M., Abreu, N.M., Netto, P.O.B., Hahn, P. and Querido, T., (2007), "A survey of the quadratic assignment problem", European Journal of Operational Research, Volume 176, Issue 2, 16, Pages 657-690.

[13] Mahdavi, I., Paydar, M.M., Solimanpur, M. and Heidarzade ,A., (2009), "Genetic algorithm approach for solving a cell formation problem in cellular manufacturing", Expert Systems with Applications 36, 6598–6604.

[14] Marti, R., Laguna, M. and Glover, F., (2006), "Principles of scatter search" , European Journal of Operational Research 169, 359-372.

[15] Misevicius, A. and Kilda, B., (2005), "The comparison of crossover operators for the quadratic assignment problem", Information Technology and Control,Vol.34,No.2.

[16] Mawdeslev, M.J. and Jibourib, S.H., (2003), "Proposed genetic algorithms for construction site layout",Engineering Applications of Artifical Intelligence 16, 501-509.

[17] Nehi, M. and Gelareh, S., (2007), "A survey of meta-heuristic solution methods for the quadratic assignment problem", Applied Mathematical Sciences,Vol.1, no.46,2293-2312.

[18] Ramkumar, A.S., Ponnambalam, S.G. and Jawahar, N., (2009), “A new iterated fast local search heuristic for solving QAP formulation in facility layout design” ,Robotics and Computer-Integrated Manufacturing 25, 620– 629.

[19] Ravindra, K.A., James, B.O. and Ashish, T., (2007) , “Agreedy genetic algorithm for the quadratic assignment problem”, Computers & Operations Research 27,917-934.

[20] Yip, P.P. and Pao, Y., (1994), “A guided evolutionary simulated annealing approach to the quadratic assignment problem”, IEEE transactions on systems, man, and cybernetics,vol.24,no:9.

[21] Yuan, P., Hu, Y.,Liu H. and Gao, H., (2003), “Scatter search algorithm for multi-headed Mounter”, EBSCO Publishing.

6. APPENDIX

6.1 Simulated Annealing – Full Tabulation

6.2 Genetic Algorithm – Full Tabulation

6.3 Scatter Search Algorithm – Full Tabulation

6.4 Grasp – Full Tabulation

6.1. Simulated Annealing – Full Tabulation

Table 6.1 Simulated Annealing – full test results

Pro	Prob.	1	2	3	4	5	6	7	8	9	10	Avg.	Best
H5	PS11AK08-9	840	774	812	810	800	800	784	788	784	798	799	774
H5	PS11AK1011	802	820	928	892	796	814	836	868	814	886	845,6	796
H5	PS11AK12-7	878	834	904	886	812	854	870	782	868	814	850,2	782
H5	PS11AK15-4	784	792	738	762	728	738	770	742	772	718	754,4	718
H5	PS11AK16-3	1380	1418	1398	1372	1392	1476	1432	1408	1480	1426	1418,2	1372
H5	PS11AK16-4	1490	1504	1490	1478	1482	1554	1468	1486	1540	1488	1498,3	1468
H5	PS11AK16-5	1532	1528	1624	1572	1610	1514	1636	1658	1494	1584	1575,2	1494
H5	PS11AK17N3	1116	1154	1156	1168	1186	1234	1176	1162	1234	1154	1174	1116
H6	PS11AK08-9	836	868	818	894	846	826	822	796	818	748	827,2	748
H6	PS11AK1011	794	874	778	842	820	810	848	798	834	806	820,4	778
H6	PS11AK12-7	848	886	806	852	874	836	916	826	882	872	859,8	806
H5	PS11AK15-4	772	714	762	760	758	730	748	778	740	738	750	714
H6	PS11AK16-3	1382	1482	1470	1368	1398	1428	1398	1404	1468	1410	1420,8	1368
H6	PS11AK16-4	1436	1432	1480	1510	1554	1506	1512	1462	1392	1466	1475	1392
H6	PS11AK16-5	1576	1670	1620	1612	1622	1572	1562	1538	1454	1562	1578,8	1454
H6	PS11AK17N3	1150	1228	1206	1214	1208	1112	1226	1198	1190	1128	1186	1112
H7	PS11AK08-9	810	840	834	858	828	774	812	808	882	774	822	774
H7	PS11AK1011	840	866	796	906	824	854	822	820	890	784	840,2	784
H7	PS11AK12-7	838	812	782	840	836	814	900	844	830	808	830,4	782
H7	PS11AK15-4	722	734	764	726	764	768	738	728	762	762	746,8	722
H7	PS11AK16-3	1462	1382	1400	1444	1400	1406	1368	1456	1392	1426	1413,6	1368
H7	PS11AK16-4	1498	1446	1500	1468	1480	1500	1456	1536	1446	1506	1483,6	1446
H7	PS11AK16-5	1558	1644	1596	1558	1540	1634	1652	1518	1582	1668	1595	1518
H7	PS11AK17N3	1180	1162	1130	1190	1194	1126	1112	1080	1198	1190	1156,2	1080
H8	PS11AK08-9	794	778	806	768	806	798	842	844	810	788	803,4	768
H8	PS11AK1011	776	832	822	850	806	822	812	768	802	844	813,4	768
H8	PS11AK12-7	802	808	830	850	868	850	814	784	868	812	828,6	784
H8	PS11AK15-4	740	750	722	770	788	760	742	722	732	762	748,8	722
H8	PS11AK16-3	1358	1376	1392	1404	1490	1424	1384	1430	1398	1358	1401,4	1358
H8	PS11AK16-4	1492	1474	1464	1540	1424	1458	1488	1522	1458	1454	1477,4	1424
H8	PS11AK16-5	1604	1720	1562	1540	1604	1492	1562	1570	1584	1598	1583,6	1492
H8	PS11AK17N3	1132	1226	1146	1142	1206	1204	1174	1128	1124	1134	1161,6	1124
H9	PS11AK08-9	858	836	860	872	884	878	872	862	890	902	871,4	836
H9	PS11AK1011	924	894	866	852	860	884	846	874	896	878	877,4	846
H9	PS11AK12-7	920	922	900	898	862	890	880	936	934	866	900,8	862
H9	PS11AK15-4	818	810	822	812	808	832	830	804	816	816	816,8	804
H9	PS11AK16-3	1462	1450	1458	1472	1446	1452	1460	1492	1454	1404	1455	1404
H9	PS11AK16-4	1486	1510	1506	1520	1464	1478	1528	1468	1508	1508	1497,6	1464
H9	PS11AK16-5	1560	1538	1542	1530	1544	1540	1550	1560	1540	1572	1547,6	1530
H9	PS11AK17N3	1182	1172	1210	1198	1206	1168	1238	1188	1190	1190	1194,2	1168
H10	PS11AK08-9	806	784	786	794	828	798	828	816	820	814	807,4	784
H10	PS11AK1011	806	812	828	800	814	814	806	822	808	816	812,6	800
H10	PS11AK12-7	814	814	800	830	842	838	842	874	844	814	831,2	800
H10	PS11AK15-4	774	780	764	764	760	786	768	776	754	768	769,4	754
H10	PS11AK16-3	1394	1420	1406	1416	1406	1410	1438	1408	1384	1398	1408	1384
H10	PS11AK16-4	1452	1422	1444	1468	1436	1446	1436	1444	1434	1434	1441,6	1422
H10	PS11AK16-5	1506	1512	1496	1494	1508	1492	1498	1528	1502	1486	1502,2	1486
H10	PS11AK17N3	1152	1140	1168	1192	1154	1160	1150	1126	1198	1180	1162	1126

H11	PS11AK08-9	802	844	834	856	798	816	874	864	824	860	837,2	798
H11	PS11AK1011	848	796	872	848	886	842	826	764	814	818	831,4	764
H11	PS11AK12-7	922	862	852	900	844	892	824	818	840	812	856,6	812
H11	PS11AK15-4	714	766	742	806	782	744	766	802	758	760	764	714
H11	PS11AK16-3	1390	1414	1378	1378	1448	1406	1372	1362	1406	1434	1398,8	1362
H11	PS11AK16-4	1540	1506	1500	1534	1458	1394	1408	1602	1492	1468	1490,2	1394
H11	PS11AK16-5	1612	1548	1514	1556	1570	1618	1588	1630	1622	1536	1579,4	1514
H11	PS11AK17N3	1146	1184	1230	1180	1232	1188	1192	1228	1120	1098	1179,8	1098
H12	PS11AK08-9	862	828	842	890	808	782	808	820	800	806	824,6	782
H12	PS11AK1011	810	820	854	826	792	818	796	802	808	758	808,4	758
H12	PS11AK12-7	874	842	896	860	846	798	834	854	808	898	851	798
H12	PS11AK15-4	772	760	740	764	774	768	818	766	772	754	768,8	740
H12	PS11AK16-3	1394	1380	1466	1378	1420	1416	1472	1348	1472	1396	1414,2	1348
H12	PS11AK16-4	1486	1502	1536	1436	1482	1408	1426	1502	1450	1458	1468,6	1408
H12	PS11AK16-5	1586	1534	1550	1566	1644	1536	1488	1546	1568	1628	1564,6	1488
H12	PS11AK17N3	1196	1156	1192	1224	1146	1160	1134	1120	1232	1202	1176,2	1120
H13	PS11AK08-9	834	900	806	816	828	838	804	824	804	830	828,4	804
H13	PS11AK1011	750	806	850	766	818	800	838	834	806	830	809,8	750
H13	PS11AK12-7	774	860	852	830	804	810	848	806	820	782	818,6	774
H13	PS11AK15-4	734	762	818	758	726	774	766	792	756	716	760,2	716
H13	PS11AK16-3	1458	1378	1396	1412	1384	1380	1404	1414	1366	1508	1410	1366
H13	PS11AK16-4	1538	1436	1456	1462	1424	1452	1486	1478	1406	1446	1458,4	1406
H13	PS11AK16-5	1620	1570	1598	1626	1486	1560	1582	1468	1646	1542	1569,8	1468
H13	PS11AK17N3	1218	1186	1232	1144	1120	1182	1180	1136	1156	1150	1170,4	1120
H14	PS11AK08-9	836	808	778	740	824	882	834	812	860	826	820	740
H14	PS11AK1011	896	790	810	842	790	756	764	798	766	854	806,6	756
H14	PS11AK12-7	848	878	800	810	870	820	800	884	912	800	842,2	800
H14	PS11AK15-4	762	752	740	768	766	718	766	766	756	726	752	718
H14	PS11AK16-3	1378	1370	1400	1398	1374	1438	1370	1388	1380	1460	1395,6	1370
H14	PS11AK16-4	1444	1434	1440	1500	1392	1494	1506	1464	1402	1536	1461,2	1392
H14	PS11AK16-5	1474	1628	1570	1482	1680	1526	1518	1554	1518	1476	1542,6	1474
H14	PS11AK17N3	1206	1180	1166	1190	1188	1076	1150	1184	1124	1190	1165,4	1076
H15	PS11AK08-9	860	846	888	886	892	890	858	870	880	896	876,6	846
H15	PS11AK1011	870	838	886	880	882	872	886	830	886	910	874	830
H15	PS11AK12-7	910	920	878	886	896	886	898	912	892	876	895,4	876
H15	PS11AK15-4	812	818	804	814	806	816	790	814	844	810	812,8	790
H15	PS11AK16-3	1488	1478	1460	1434	1476	1466	1468	1498	1440	1460	1466,8	1434
H15	PS11AK16-4	1472	1572	1478	1484	1496	1490	1522	1508	1482	1562	1506,6	1472
H15	PS11AK16-5	1536	1528	1544	1514	1562	1518	1538	1512	1496	1524	1527,2	1496
H15	PS11AK17N3	1226	1210	1170	1222	1226	1192	1218	1254	1192	1246	1215,6	1170
H16	PS11AK08-9	804	818	806	812	794	798	788	792	804	860	807,6	788
H16	PS11AK1011	816	808	836	802	810	816	810	836	812	826	817,2	802
H16	PS11AK12-7	872	830	840	818	832	828	850	836	836	854	839,6	818
H16	PS11AK15-4	760	768	780	778	774	770	774	762	754	770	769	754
H16	PS11AK16-3	1396	1408	1404	1422	1408	1416	1412	1420	1406	1418	1411	1396
H16	PS11AK16-4	1454	1450	1452	1438	1444	1482	1446	1444	1494	1442	1454,6	1438
H16	PS11AK16-5	1514	1506	1524	1482	1498	1514	1494	1508	1532	1512	1508,4	1482
H16	PS11AK17N3	1186	1150	1184	1172	1154	1148	1162	1140	1135	1154	1158,5	1135

6.2. Genetic Algorithm – Full Tabulation

Table 6.2 Genetic Algorithm – full test results

Pro	Prob.	1	2	3	4	5	6	7	8	9	10	Avg.	Best
G1	PS11AK08-9	1066	1080	1132	1132	1090	1048	1004	1156	1126	1074	1090,8	1004
G1	PS11AK1011	1024	968	1206	954	1072	1032	1136	994	982	1010	1037,8	954
G1	PS11AK12-7	986	1182	1096	1062	1180	1064	1050	1050	1040	1176	1088,6	986
G1	PS11AK15-4	892	856	864	870	890	854	882	940	890	854	879,2	854
G1	PS11AK16-3	1732	1692	1630	1708	1520	1714	1748	1622	1616	1718	1670	1520
G1	PS11AK16-4	1650	1634	1922	1684	1620	1812	1732	1830	1750	1668	1730,2	1620
G1	PS11AK16-5	1830	2114	1948	1992	1860	1852	1844	2064	1880	1922	1930,6	1830
G1	PS11AK17N3	1502	1720	1632	1634	1700	1608	1496	1634	1580	1566	1607,2	1496
G2	PS11AK08-9	906	962	978	982	944	942	1006	956	964	918	955,8	906
G2	PS11AK1011	1046	928	986	1006	910	968	922	966	920	928	958	910
G2	PS11AK12-7	1024	1056	962	990	954	982	1006	1084	1000	952	1001	952
G2	PS11AK15-4	830	834	836	792	850	810	824	866	786	822	825	786
G2	PS11AK16-3	1600	1488	1626	1434	1676	1504	1526	1506	1508	1510	1537,8	1434
G2	PS11AK16-4	1546	1632	1608	1612	1646	1704	1638	1530	1652	1596	1616,4	1530
G2	PS11AK16-5	1788	1680	1872	1698	1714	1648	1878	1912	1734	1756	1768	1648
G2	PS11AK17N3	1420	1404	1524	1338	1492	1454	1300	1362	1354	1414	1406,2	1300
G3	PS11AK08-9	958	934	964	924	920	930	936	888	886	872	921,2	872
G3	PS11AK1011	900	958	952	870	900	970	888	967	870	890	916,5	870
G3	PS11AK12-7	952	874	996	1000	956	954	1022	970	946	960	963	874
G3	PS11AK15-4	746	786	802	748	762	752	798	780	776	794	774,4	746
G3	PS11AK16-3	1498	1464	1492	1454	1464	1450	1478	1478	1434	1478	1469	1434
G3	PS11AK16-4	1528	1546	1536	1574	1556	1520	1538	1572	1564	1520	1545,4	1520
G3	PS11AK16-5	1698	1580	1722	1700	1678	1704	1620	1668	1698	1670	1673,8	1580
G3	PS11AK17N3	1410	1296	1292	1344	1298	1418	1264	1286	1456	1406	1347	1264
G4	PS11AK08-9	924	924	898	888	892	906	860	882	908	922	900,4	860
G4	PS11AK1011	892	948	944	850	870	900	888	958	848	880	897,8	848
G4	PS11AK12-7	954	968	972	934	936	1006	978	894	978	898	951,8	894
G4	PS11AK15-4	772	852	844	792	766	826	776	824	836	780	806,8	766
G4	PS11AK16-3	1484	1536	1422	1430	1448	1498	1454	1500	1510	1446	1472,8	1422
G4	PS11AK16-4	1586	1544	1478	1540	1636	1488	1538	1504	1626	1504	1544,4	1478
G4	PS11AK16-5	1640	1714	1734	1710	1702	1672	1550	1582	1730	1616	1665	1550
G4	PS11AK17N3	1402	1366	1338	1288	1344	1314	1246	1308	1358	1334	1329,8	1246
G5	PS11AK08-9	956	928	868	868	934	904	858	886	928	910	904	858
G5	PS11AK1011	812	818	964	802	882	874	932	846	902	912	874,4	802
G5	PS11AK12-7	978	942	934	926	912	854	964	916	862	942	923	854
G5	PS11AK15-4	748	798	736	786	736	764	802	804	748	762	768,4	736
G5	PS11AK16-3	1416	1390	1452	1444	1412	1432	1424	1414	1428	1392	1420,4	1390
G5	PS11AK16-4	1532	1510	1498	1538	1510	1568	1554	1490	1498	1458	1515,6	1458
G5	PS11AK16-5	1612	1600	1530	1632	1724	1598	1598	1520	1610	1514	1593,8	1514
G5	PS11AK17N3	1272	1300	1232	1306	1294	1262	1314	1260	1374	1196	1281	1196
G6	PS11AK08-9	850	838	868	876	842	850	840	876	806	916	856,2	806
G6	PS11AK1011	864	840	830	880	810	866	808	846	900	848	849,2	808
G6	PS11AK12-7	908	914	926	920	948	950	860	958	930	858	917,2	858
G6	PS11AK15-4	746	744	792	776	776	744	716	798	782	754	762,8	716
G6	PS11AK16-3	1456	1408	1378	1418	1388	1430	1422	1382	1408	1386	1407,6	1378
G6	PS11AK16-4	1466	1514	1476	1516	1458	1494	1478	1504	1528	1472	1490,6	1458
G6	PS11AK16-5	1666	1546	1638	1564	1602	1612	1632	1588	1642	1548	1603,8	1546
G6	PS11AK17N3	1220	1154	1268	1202	1242	1208	1302	1236	1188	1284	1230,4	1154
G7	PS11AK08-9	1268	1280	1336	1274	1310	1360	1402	1430	1292	1372	1332,4	1268
G7	PS11AK1011	1272	1326	1296	1192	1236	1370	1350	1332	1426	1220	1302	1192

G7	PS11AK12-7	1472	1240	1398	1452	1384	1302	1300	1410	1372	1484	1381,4	1240
G7	PS11AK15-4	960	1096	1062	1006	916	1020	1014	1056	980	984	1009,4	916
G7	PS11AK16-3	1938	1964	2088	1874	2000	1866	2234	1836	2068	2088	1995,6	1836
G7	PS11AK16-4	2100	1922	1948	1938	2272	2142	2238	1926	2092	1920	2049,8	1920
G7	PS11AK16-5	2096	2312	2460	2302	2262	2150	2258	2160	2314	2408	2272,2	2096
G7	PS11AK17N3	1812	2040	2090	1840	1992	1746	1862	1764	1932	1810	1888,8	1746
G8	PS11AK08-9	1370	1338	1186	1252	1226	1366	1316	1220	1290	1220	1268,22	1186
G8	PS11AK1011	1146	1274	1246	1208	1172	1262	1194	1376	1210	1176	1235,33	1172
G8	PS11AK12-7	1366	1216	1484	1404	1370	1264	1294	1250	1402	1290	1330,44	1216
G8	PS11AK15-4	966	978	1056	1076	934	1082	1060	1024	1048	994	1028	934
G8	PS11AK16-3	2036	1916	2036	1926	1912	1838	2078	2006	1838	1964	1946	1838
G8	PS11AK16-4	2284	1958	1972	2092	2058	2034	2204	2004	2000	2184	2056,22	1958
G8	PS11AK16-5	2208	2098	2158	2176	2206	2260	2372	2278	2184	2220	2216,88	2098
G8	PS11AK17N3	1814	1788	1724	1846	1832	1964	1920	1844	1676	1696	1810	1676
G9	PS11AK08-9	1358	1274	1240	1356	1376	1278	1240	1298	1266	1208	1289,4	1208
G9	PS11AK1011	1238	1372	1276	1200	1332	1192	1318	1278	1194	1110	1251	1110
G9	PS11AK12-7	1286	1420	1396	1308	1344	1266	1282	1330	1318	1246	1319,6	1246
G9	PS11AK15-4	1044	1020	1032	956	984	1024	1030	1004	994	984	1007,2	956
G9	PS11AK16-3	1924	1954	1946	2032	2034	1998	1890	2150	1934	1874	1973,6	1874
G9	PS11AK16-4	1956	2236	2080	2098	1944	1950	1846	2150	1948	2094	2038,44	1846
G9	PS11AK16-5	2132	2238	2152	2258	2144	2264	2332	2158	2282	2172	2222,22	2144
G9	PS11AK17N3	1908	1738	1850	1774	1810	1776	1774	1850	1720	1880	1796,88	1720
G10	PS11AK08-9	1092	1034	1070	946	1058	1118	960	976	1076	1060	1039	946
G10	PS11AK1011	960	1170	1200	1002	1026	938	1016	990	1100	980	1038,2	938
G10	PS11AK12-7	998	1078	998	1026	1010	1134	1052	1098	1110	964	1046,8	964
G10	PS11AK15-4	832	872	864	856	866	866	848	838	874	838	855,4	832
G10	PS11AK16-3	1736	1700	1608	1636	1524	1700	1598	1826	1660	1628	1661,6	1524
G10	PS11AK16-4	1640	1646	1612	1750	1730	1698	1766	1666	1680	1636	1682,4	1612
G10	PS11AK16-5	1822	1828	1892	1828	1960	1884	1792	1722	1898	1686	1831,2	1686
G10	PS11AK17N3	1484	1528	1390	1444	1508	1442	1516	1556	1436	1424	1472,8	1390
G11	PS11AK08-9	1054	916	1018	952	1076	1032	980	974	948	1042	999,2	916
G11	PS11AK1011	918	1040	940	992	1012	916	1018	910	1086	902	973,4	902
G11	PS11AK12-7	1016	1018	954	1014	940	1136	1110	1004	1038	1034	1026,4	940
G11	PS11AK15-4	814	852	816	830	818	792	806	842	866	838	827,4	792
G11	PS11AK16-3	1506	1648	1562	1508	1628	1598	1616	1578	1536	1586	1576,6	1506
G11	PS11AK16-4	1648	1634	1642	1662	1548	1644	1672	1606	1582	1704	1634,2	1548
G11	PS11AK16-5	1794	1716	1852	1748	1858	1704	1800	1686	1814	1898	1787	1686
G11	PS11AK17N3	1424	1406	1442	1422	1362	1572	1470	1446	1386	1506	1443,6	1362
G12	PS11AK08-9	984	868	968	964	914	952	998	1008	946	932	953,4	868
G12	PS11AK1011	1000	998	952	904	906	942	876	970	966	972	948,6	876
G12	PS11AK12-7	938	966	978	974	954	910	998	980	980	1016	969,4	910
G12	PS11AK15-4	810	808	796	808	828	842	836	798	800	776	810,2	776
G12	PS11AK16-3	1556	1504	1510	1554	1642	1466	1558	1530	1630	1552	1550,2	1466
G12	PS11AK16-4	1654	1620	1596	1618	1674	1598	1612	1612	1630	1716	1633	1596
G12	PS11AK16-5	1682	1790	1756	1868	1826	1750	1716	1868	1816	1764	1783,6	1682
G12	PS11AK17N3	1352	1330	1346	1310	1366	1418	1390	1310	1420	1346	1358,8	1310

6.3. Scatter Search – Full Tabulation

Table 6.3 Scatter Search Algorithm – full test results

Pro	Prob.	1	2	3	4	5	6	7	8	9	10	Avg.	Best
S1	PS11AK08-9	1358	1190	932	1324	1354	1304	1238	1200	1272	1280	1245,2	932
S1	PS11AK1011	1202	1170	1292	1322	1448	1138	1210	1404	1418	1382	1298,6	1138
S1	PS11AK12-7	1340	1326	1350	1478	1438	1294	1548	1174	1462	1434	1384,4	1174
S1	PS11AK15-4	1138	948	896	972	1042	952	1058	1060	1122	962	1015,0	896
S1	PS11AK16-3	1900	2244	1902	2186	2040	1968	2090	1894	2078	2318	2062	1894
S1	PS11AK16-4	2130	2076	1990	2212	2676	2226	1970	1906	2060	1986	2132,2	1906
S1	PS11AK16-5	2088	2544	2306	2610	2050	2034	1944	1974	2366	2318	2223,4	1944
S1	PS11AK17N3	1482	1730	1726	2072	1800	1766	1792	2260	1786	1752	1816,6	1482
S2	PS11AK08-9	1170	1082	1116	1388	1246	1326	1276	1116	1186	1094	1200	1082
S2	PS11AK1011	1086	1244	1194	1272	1180	1496	1282	1204	1112	1466	1253,6	1086
S2	PS11AK12-7	1236	1302	1070	1144	1076	1370	1330	1180	1464	1442	1261,4	1070
S2	PS11AK15-4	1186	860	1012	880	1030	1068	1162	1134	1170	1018	1052	860
S2	PS11AK16-3	2478	2126	1756	1902	1928	2244	2010	1850	1814	2188	2029,6	1756
S2	PS11AK16-4	2176	1812	2036	1948	1988	1736	2096	2180	1686	1872	1953	1686
S2	PS11AK16-5	2586	2268	2130	2376	2538	2258	2444	2256	2272	2304	2343,2	2130
S2	PS11AK17N3	1894	1596	1824	1896	1728	1716	2060	1990	2108	1544	1835,6	1544
S3	PS11AK08-9	1082	1222	1172	1386	1046	1090	1300	1186	1222	958	1166,4	958
S3	PS11AK1011	1022	1078	1466	1240	990	1278	1126	1298	1192	1122	1181,2	990
S3	PS11AK12-7	1376	1256	1132	1010	1362	1256	1064	1270	1126	1126	1197,8	1010
S3	PS11AK15-4	966	824	938	1046	984	1026	888	844	1006	1086	960,8	824
S3	PS11AK16-3	1998	1540	1766	1646	2086	2018	1284	1784	1676	1444	1724,2	1284
S3	PS11AK16-4	2042	2220	1708	2016	1868	1866	2034	1922	1756	1962	1939,4	1708
S3	PS11AK16-5	1726	2004	1734	1800	1790	2144	2428	1718	2042	2192	1957,8	1718
S3	PS11AK17N3	1606	1380	1778	1540	1550	2018	1710	1776	1430	1918	1670,6	1380
S4	PS11AK08-9	956	1176	916	1156	1210	1334	1092	992	950	1266	1104,8	916
S4	PS11AK1011	1042	1004	1076	1178	1160	1166	1156	1162	1378	936	1125,8	936
S4	PS11AK12-7	1148	984	1192	1190	1092	1208	1394	1136	1098	1320	1176,2	984
S4	PS11AK15-4	792	972	768	842	914	894	926	1130	932	870	904	768
S4	PS11AK16-3	1902	1968	1538	1588	1630	1928	1534	1514	1762	1572	1693,6	1514
S4	PS11AK16-4	1812	1800	1910	1592	2020	2030	1960	1898	1552	1694	1826,8	1552
S4	PS11AK16-5	2316	1988	2020	2010	2164	1850	2100	1842	1808	1656	1975,4	1656
S4	PS11AK17N3	1472	1862	1620	1776	1378	1962	1954	1526	1828	1338	1671,6	1338
S5	PS11AK08-9	984	1008	1056	1214	950	900	870	1146	912	892	993,2	870
S5	PS11AK1011	858	968	992	1180	872	1070	872	1150	930	874	976,6	858
S5	PS11AK12-7	932	1276	922	986	1054	1120	1154	1004	954	886	1028,8	886
S5	PS11AK15-4	862	894	790	822	902	772	814	832	824	846	853,8	772
S5	PS11AK16-3	1520	1848	1540	1774	1842	1620	1614	1398	1554	1436	1614,6	1398
S5	PS11AK16-4	1934	1574	1690	1560	1738	2036	1534	1502	1602	1672	1684,2	1502
S5	PS11AK16-5	1540	2202	1838	1748	1660	1666	1814	1904	1946	1708	1802,6	1540
S5	PS11AK17N3	1666	1648	1636	1248	1848	1324	1288	1472	1410	1400	1494	1248
S6	PS11AK08-9	1240	1108	1084	1148	1072	1188	1074	1024	1116	1014	1106,8	1014
S6	PS11AK1011	1254	1182	1166	1244	1068	1304	1188	1396	1072	1292	1216,6	1068
S6	PS11AK12-7	1534	1252	1250	1542	1262	1298	1486	1486	1190	1164	1346,4	1164
S6	PS11AK15-4	1076	896	992	890	944	994	1142	918	986	1016	985,4	890

S6	PS11AK16-3	1824	1946	1994	1916	1818	1756	1612	2118	1902	1888	1877,4	1612
S6	PS11AK16-4	1944	2018	1626	2234	2094	2046	1904	2170	2178	2056	2027	1626
S6	PS11AK16-5	1916	2118	2324	1864	2322	2328	2534	2328	2174	2124	2203,2	1864
S6	PS11AK17N3	2270	1660	2144	1618	1694	1876	1954	1876	1920	1602	1861,4	1602
S7	PS11AK08-9	890	928	912	918	1092	989	936	1042	1256	1014	977,7	890
S7	PS11AK1011	1100	908	934	1230	904	1062	950	1034	1058	870	1005	870
S7	PS11AK12-7	1094	988	1210	1274	974	956	1018	1124	1226	1220	1108,4	956
S7	PS11AK15-4	950	1112	890	918	986	1016	1024	996	930	970	979,2	890
S7	PS11AK16-3	1922	1812	1606	1434	1528	1888	1920	1750	1800	1820	1748	1434
S7	PS11AK16-4	1498	1860	1450	1616	1546	1520	1820	1560	1592	1530	1599,2	1450
S7	PS11AK16-5	1790	1904	2082	1708	2078	1634	2316	1662	2110	2000	1928,4	1634
S7	PS11AK17N3	1362	1364	1332	1396	1438	1418	1496	1760	1680	1650	1489,6	1332
S8	PS11AK08-9	986	1224	1056	926	940	1352	1078	1006	1115	1058	1074,1	926
S8	PS11AK1011	1054	1338	1116	1204	1042	1080	1438	1080	1054	934	1134	934
S8	PS11AK12-7	1004	1126	1080	1120	1306	1170	1040	1060	1420	1160	1148,6	1004
S8	PS11AK15-4	986	976	964	836	918	922	1142	818	832	930	932,4	818
S8	PS11AK16-3	1924	1638	2130	1776	1770	1822	1772	1562	2114	2126	1863,4	1562
S8	PS11AK16-4	1642	1720	1896	1904	1798	1964	1972	1806	1660	1756	1811,8	1642
S8	PS11AK16-5	1900	2228	2276	1972	2228	1850	1960	1932	1990	1846	2018,2	1846
S8	PS11AK17N3	1496	1704	1734	1860	1882	1850	1800	1758	1512	1812	1740,8	1496
S9	PS11AK08-9	1184	1242	1188	998	1208	946	888	1170	1108	1496	1142,8	888
S9	PS11AK1011	1522	1126	1142	992	1142	1074	1272	1330	1054	1292	1194,6	992
S9	PS11AK12-7	1612	1722	1728	1644	1574	1606	1510	1520	1476	1504	1589,6	1476
S9	PS11AK15-4	1154	1096	1068	1200	1108	1154	1162	1158	1110	1212	1142,2	1068
S9	PS11AK16-3	2264	2206	2112	2216	2102	2150	2078	1932	2246	2208	2151,4	1932
S9	PS11AK16-4	1614	1942	1900	1902	1542	1726	1756	1626	1924	1814	1774,6	1542
S9	PS11AK16-5	1892	2004	2114	1738	2126	2032	2120	1952	1844	1886	1970,8	1738
S9	PS11AK17N3	1688	1480	1750	1320	1640	1908	1392	1324	2070	1472	1604,4	1320
S10	PS11AK08-9	1770	1490	1690	1626	1644	1664	1614	1686	1682	1496	1636,2	1490
S10	PS11AK1011	1548	1624	1678	1538	1726	1592	1662	1606	1608	1620	1620,2	1538
S10	PS11AK12-7	1540	1516	1900	1750	1714	1730	1724	1656	1702	1698	1693	1516
S10	PS11AK15-4	1278	1190	1232	1264	1244	1210	1240	1254	1224	1194	1233	1190
S10	PS11AK16-3	2062	2348	2356	2290	2146	2348	2394	2428	2286	2388	2304,6	2062
S10	PS11AK16-4	2334	2408	2330	2388	2398	2264	2184	2444	2238	2576	2356,4	2184
S10	PS11AK16-5	2700	2706	2742	2554	2656	2732	2560	2806	2558	2674	2668,8	2554
S10	PS11AK17N3	2246	2220	2142	2232	2292	2214	2174	2212	2302	2398	2243,2	2142
S11	PS11AK08-9	1612	1686	1702	1644	1562	1670	1640	1658	1664	1570	1640,8	1562
S11	PS11AK1011	1688	1656	1578	1662	1612	1546	1606	1540	1616	1524	1602,8	1524
S11	PS11AK12-7	1846	1734	1762	1712	1832	1868	1776	1706	1728	1752	1772,6	1706
S11	PS11AK15-4	1140	1232	1232	1344	1196	1186	1142	1178	1212	1138	1200	1138
S11	PS11AK16-3	2346	2334	2192	2408	2418	2368	2468	2252	2348	2202	2333,6	2192
S11	PS11AK16-4	2306	2442	2292	2424	2350	2490	2278	2386	2400	2278	2364,6	2278
S11	PS11AK16-5	2320	2668	2616	2716	2432	2598	2494	2592	2512	2546	2549,4	2320
S11	PS11AK17N3	2340	2234	2298	2416	2286	2142	2228	2460	2226	2308	2293,8	2142
S12	PS11AK08-9	1526	1644	1674	1474	1458	1568	1640	1678	1568	1542	1577,2	1458
S12	PS11AK1011	1514	1596	1382	1588	1684	1570	1548	1624	1562	1492	1556	1382
S12	PS11AK12-7	1786	1770	1752	1714	1718	1672	1618	1698	1672	1706	1710,6	1618
S12	PS11AK15-4	1214	1350	1030	1212	1362	1330	1252	1144	1172	1202	1266,8	1030
S12	PS11AK16-3	2306	2212	2222	2490	2324	2204	2200	2258	2344	2222	2278,2	2200
S12	PS11AK16-4	2436	2244	2318	2216	2380	2264	2378	2506	2332	2212	2328,6	2212
S12	PS11AK16-5	2620	2464	2718	2482	2462	2402	2276	2694	2474	2410	2500,2	2276
S12	PS11AK17N3	2378	2222	2122	2234	2172	2114	2350	2350	2258	2368	2256,8	2214

S13	PS11AK08-9	1572	1632	1620	1512	1608	1640	1466	1466	1532	1550	1559,8	1466
S13	PS11AK1011	1544	1600	1530	1678	1524	1432	1606	1648	1530	1488	1558	1432
S13	PS11AK12-7	1628	1760	1712	1634	1538	1646	1698	1514	1696	1726	1655,2	1514
S13	PS11AK15-4	1168	1234	1226	1206	1268	1186	1168	1194	1220	1228	1209,8	1168
S13	PS11AK16-3	258	2304	2300	2128	2156	2280	2250	2124	2270	2248	2231,8	2124
S13	PS11AK16-4	2386	2180	2316	2306	2326	2416	2414	2350	2228	2374	2329,6	2180
S13	PS11AK16-5	2352	2302	2444	2542	2466	2516	2274	2546	2612	2470	2450,44	2274
S13	PS11AK17N3	2182	2152	2096	2234	2208	2160	2344	2158	2124	2212	2187	2096
S14	PS11AK08-9	1606	1606	1546	1606	1556	1630	1544	1548	1598	1566	1580,6	1544
S14	PS11AK1011	1550	1464	1608	1554	1502	1500	1526	1520	1520	1526	1527	1464
S14	PS11AK12-7	1688	1726	1606	1574	1676	1628	1698	1674	1672	1758	1670	1574
S14	PS11AK15-4	1126	1186	1180	1158	1182	1178	1082	1210	1124	1100	1152,6	1082
S14	PS11AK16-3	2316	2068	2372	2274	2132	2084	2092	2250	2254	2214	2205,6	2068
S14	PS11AK16-4	2144	2376	2286	2234	2224	2358	2244	2296	2242	2296	2270	2144
S14	PS11AK16-5	2498	2440	2560	2466	2466	2544	2562	2592	2510	2526	2516,4	2440
S14	PS11AK17N3	2168	2206	2136	2314	2130	2004	2372	2380	2188	2223	2212,1	2004

6.4. Grasp – Full Tabulation

Table 6.4 Grasp – full test results

Pro	Prob.	1	2	3	4	5	6	7	8	9	10	Avg.	Best
R1	PS11AK08-9	1564	1576	1546	1612	1644	1626	1592	1560	1564	1566	1.585,0	1546
R1	PS11AK1011	1440	1634	1468	1530	1578	1576	1448	1466	1576	1516	1.523,2	1440
R1	PS11AK12-7	1602	1608	1578	1588	1522	1582	1536	1626	1636	1600	1.587,8	1522
R1	PS11AK15-4	1166	1228	1284	1296	1276	1304	1162	1240	1246	1300	1.250,2	1162
R1	PS11AK16-3	2350	2174	2422	2262	2372	2332	2414	2270	2364	2294	2.325,4	2174
R1	PS11AK16-4	2574	2382	2272	2380	2382	2426	2336	2346	2418	2500	2.401,6	2272
R1	PS11AK16-5	2446	2660	2650	2356	2568	2536	2558	2620	2626	2534	2.555,4	2356
R1	PS11AK17N3	2352	2354	2416	2450	2226	2376	2406	2462	2272	2442	2.375,6	2226
R2	PS11AK08-9	1282	1190	1256	1132	1238	1370	1264	1168	1246	1206	1.235,2	1132
R2	PS11AK1011	1284	1228	1212	1330	1250	1172	1248	1204	1306	1328	1.256,2	1172
R2	PS11AK12-7	1382	1308	1326	1332	1242	1524	1420	1292	1438	1396	1.366,0	1242
R2	PS11AK15-4	836	962	1016	986	958	974	938	1008	1028	910	961,6	836
R2	PS11AK16-3	2020	1926	1812	1802	1986	1848	1914	1792	1942	1894	1.893,6	1792
R2	PS11AK16-4	1954	2028	2010	1998	1940	1904	1912	1924	1938	1980	1.958,8	1904
R2	PS11AK16-5	2164	2262	2000	2218	2054	2016	2128	2024	2006	2192	2.106,4	2000
R2	PS11AK17N3	2058	1828	1928	1784	1944	1782	1626	1928	2064	1922	1.886,4	1626
R3	PS11AK08-9	938	882	966	968	902	886	930	1020	946	1024	946,2	882
R3	PS11AK1011	966	946	946	910	904	970	956	952	902	914	936,6	902
R3	PS11AK12-7	992	962	1062	1020	884	954	1088	994	976	1052	998,4	884
R3	PS11AK15-4	816	768	816	812	812	800	904	858	814	820	822,0	768
R3	PS11AK16-3	1504	1532	1528	1520	1520	1574	1444	1492	1576	1590	1.528,0	1444
R3	PS11AK16-4	1554	1640	1584	1618	1690	1578	1588	1612	1560	1600	1.602,4	1554
R3	PS11AK16-5	1684	1736	1742	1628	1738	1660	1760	1714	1776	1694	1.713,2	1628
R3	PS11AK17N3	1380	1336	1430	1416	1402	1426	1432	1412	1436	1314	1.398,4	1314
R4	PS11AK08-9	1004	916	980	986	964	940	968	904	1026	986	967,4	904
R4	PS11AK1011	886	926	872	944	890	924	976	956	1044	976	939,4	872
R4	PS11AK12-7	928	974	982	1010	986	970	1042	948	1006	982	982,8	928
R4	PS11AK15-4	826	842	792	772	840	772	742	846	860	774	806,6	742
R4	PS11AK16-3	1540	1476	1570	1486	1450	1584	1562	1508	1620	1482	1.527,8	1450
R4	PS11AK16-4	1616	1566	1510	1580	1700	1526	1534	1590	1626	1574	1.582,2	1510
R4	PS11AK16-5	1762	1632	1738	1688	1684	1668	1762	1664	1640	1634	1.687,2	1632
R4	PS11AK17N3	1458	1320	1474	1332	1448	1354	1390	1492	1338	1422	1.402,8	1320
R5	PS11AK08-9	874	1000	884	980	904	984	940	950	994	882	939,2	874
R5	PS11AK1011	908	898	964	956	888	932	1048	956	960	972	948,2	888
R5	PS11AK12-7	998	982	948	950	1002	1018	1046	1088	886	1102	1.002,0	886
R5	PS11AK15-4	806	814	846	810	806	762	830	788	798	788	804,8	762
R5	PS11AK16-3	1602	1480	1532	1528	1550	1526	1526	1584	1474	1512	1.531,4	1474
R5	PS11AK16-4	1692	1532	1558	1592	1542	1562	1672	1488	1594	1486	1.571,8	1486
R5	PS11AK16-5	1820	1556	1804	1636	1690	1760	1810	1714	1608	1624	1.702,2	1556
R5	PS11AK17N3	1486	1484	1400	1418	1332	1346	1406	1332	1400	1454	1.405,8	1332
R6	PS11AK08-9	1068	1090	1076	1130	1036	1128	1160	1044	1294	1088	1.111,4	1036
R6	PS11AK1011	1060	1014	1020	1124	1092	1052	1002	1190	1018	1102	1.067,4	1002
R6	PS11AK12-7	1222	1158	1272	1220	1212	1246	1072	1226	1164	1178	1.197,0	1072
R6	PS11AK15-4	868	902	890	900	944	868	828	876	890	874	884,0	828

R6	PS11AK16-3	1898	1682	1696	1662	1602	1578	1660	1612	1762	1678	1.683,0	1578
R6	PS11AK16-4	1674	1786	1776	1864	1916	1758	1770	1770	1816	1774	1.790,4	1674
R6	PS11AK16-5	1880	1868	1898	1920	1982	1936	1946	1912	1844	1816	1.900,2	1816
R6	PS11AK17N3	1580	1764	1634	1748	1660	1718	1726	1566	1610	1486	1.649,2	1486
R7	PS11AK08-9	1028	1106	1040	1046	1186	1010	1084	996	1134	1138	1.076,8	996
R7	PS11AK1011	1066	1212	1022	1016	1044	1170	1100	1098	1036	1072	1.083,6	1016
R7	PS11AK12-7	1102	1176	1160	1168	1004	1238	1102	1200	1126	1102	1.137,8	1004
R7	PS11AK15-4	894	866	866	876	818	900	856	870	942	880	876,8	818
R7	PS11AK16-3	1924	1616	1668	1794	1754	1602	1604	1650	1620	1600	1.683,2	1600
R7	PS11AK16-4	1674	1744	1662	1674	1788	1704	1688	1742	1828	1676	1.718,0	1662
R7	PS11AK16-5	1902	2100	1838	1972	1912	1886	2000	1950	1824	1950	1.933,4	1824
R7	PS11AK17N3	1592	1618	1682	1638	1730	1662	1732	1548	1616	1776	1.659,4	1548
R8	PS11AK08-9	1144	1222	1200	1058	1072	1104	1296	1246	1134	1120	1.159,6	1058
R8	PS11AK1011	1076	1066	1048	1064	1236	1184	1146	1058	1030	1006	1.091,4	1006
R8	PS11AK12-7	1220	1206	1214	1196	1204	1168	1280	1204	1216	1142	1.205,0	1142
R8	PS11AK15-4	932	914	954	942	874	880	904	896	846	866	900,8	846
R8	PS11AK16-3	1798	1820	1698	1556	1698	1866	1748	1684	1802	1730	1.740,0	1556
R8	PS11AK16-4	1646	1694	1674	1972	1722	1840	1782	1862	1840	1770	1.780,2	1646
R8	PS11AK16-5	1976	1940	1934	1914	1868	2032	2108	1830	1844	2004	1.945,0	1830
R8	PS11AK17N3	1640	1620	1710	1564	1504	1718	1666	1624	1552	1546	1.614,4	1504
R9	PS11AK08-9	990	982	1096	1102	1064	1180	1042	1172	1078	1082	1.078,8	982
R9	PS11AK1011	1116	1152	1010	1080	1136	978	1082	1050	956	1104	1.066,4	956
R9	PS11AK12-7	1118	1118	1024	1218	1126	1240	1170	1048	1134	1264	1.146,0	1024
R9	PS11AK15-4	778	920	820	816	894	872	838	818	864	868	848,8	778
R9	PS11AK16-3	1648	1644	1650	1654	1654	1734	1730	1564	1576	1622	1.647,6	1564
R9	PS11AK16-4	1592	1800	1770	1752	1724	1770	1702	1728	1714	1714	1.726,6	1592
R9	PS11AK16-5	2176	1932	2026	1760	1894	1884	1784	1934	2006	1926	1.932,2	1760
R9	PS11AK17N3	1570	1622	1448	1626	1516	1626	1430	1608	1620	1560	1.562,6	1430
R10	PS11AK08-9	908	1004	918	920	966	916	950	968	988	852	939,0	852
R10	PS11AK1011	1008	914	968	918	910	898	988	908	836	1028	937,6	836
R10	PS11AK12-7	976	968	1102	1018	1044	948	1054	944	1116	1044	1.021,4	944
R10	PS11AK15-4	840	826	980	846	842	850	790	824	818	850	846,6	790
R10	PS11AK16-3	1578	1494	1506	1538	1564	1542	1530	1574	1490	1612	1.542,8	1490
R10	PS11AK16-4	1560	1582	1698	1712	1614	1622	1624	1546	1510	1638	1.610,6	1510
R10	PS11AK16-5	1682	1708	1654	1648	1668	1672	1706	1788	1654	1616	1.679,6	1616
R10	PS11AK17N3	1380	1378	1406	1366	1476	1346	1408	1482	1506	1440	1.418,8	1346

7. CURRICULUM VITAE

Burcu KIYICIĞI
11/12/1983 Istanbul, Turkey
E-Mail: 200791001@dogus.edu.tr

Education

M.Sc. in Computer and Information Technology Sciences, not graduated, Dogus University (TUBITAK scholarship of my thesis), Institute of Science and Technology, Istanbul-Turkey.

B.Sc. in Computer Engineering, Dogus University, 2006, Istanbul-Turkey.