IJISAE

International Journal of
Intelligent Systems and
Applications in Engineering

ISSN:2147-6799 www.atscience.org/IJISAE

Science academic publisher
Advanced Technology and Science

Original Research Paper

A Bee Colony Optimization-based Approach for Binary Optimization

Mustafa Servet Kıran *[a], Mesut Gündüz [a]

Received 11th September 2013, Accepted 21th September 2013

Abstract: *The bee colony optimization (BCO) algorithm, one of the swarm intelligence algorithms, is a population based iterative search algorithm. Being inspired by collective bee intelligence, BCO has been proposed for solving discrete optimization problems such as travelling salesman problem. The BCO uses constructive approach for creating a feasible solution for the discrete optimization problems but in this study, we used the solution improvement technique due to nature of the uncapacitated facility location problem (UFLP). In the proposed method named as binBCO, the feasible solutions are generated for the artificial bees in hive of BCO and these solutions are tried to improve by utilizing interaction in the hive. At the end of the each iteration, some of the bees leave self-solutions and the leaving process depends on the loyalty of the bee to the self-solution. After a bee leaves self-solution, a random feasible solution is generated and assigned to this bee. In order to show the performance of binBCO, we examined it on well-known UFLPs, and the experimental studies show that the proposed method produces promising results.*

Keywords: *Bee colony optimization, Binary optimization, Uncapacitated facility location problem.*

1. Introduction

In recent years, many swarm intelligence-based optimization algorithms have been proposed in the literature, such as ant colony optimization [1,2], particle swarm optimization [3], bee colony optimization [4], artificial bee colony [5]. The BCO algorithm was proposed for solving travelling salesman problem which is a combinatorial optimization problem. In the basic BCO concepts, the partial tours are created, and while these partial tours are constructed, interactions such as information sharing, tour sharing are used. In this study, BCO, a constructive method for solving travelling salesman problem, is modified for binary optimization and transformed to path improvement method due to the fact that the objective function value of partial solutions cannot be calculated in dealing problem in this study.

Since invention of BCO in 2001, many discrete optimization problems have been resolved by this technique. The BCO was first applied to solve travelling salesman problem [6-8], and BCO is used for solving vehicle routing problem with uncertainty demand [9], ride matching problem [10,11], travelling salesman problem and a routing problem in networks [12], routing and wavelength assignment in all-optical networks [13], static scheduling of independent tasks on homogeneous multiprocessor systems [14], driver-line-time scheduling [15], p-center problem [16]. A detailed literature review about bee colony optimization and the other bee colony-based swarm intelligence techniques can be found in [4, 17].

Based on the literature review, BCO algorithm is applied to solve many discrete optimization problems. In this study, BCO algorithm is used for solving uncapacitated facility location problem by developing a binary version the BCO algorithm. The rest of paper is organized as follows. The mathematical background and objective of UFLP, BCO algorithm and proposed method are explained in the Section 2. The results of the simulations are given in Section 3 and the results are discussed in Section 4. Finally, conclusion and future works are presented in Section 5.

2. Material and Methods

2.1. Mathematical Model of Uncapacitated Facility Location Problem

A brief description about UFLP which is used to show the performance and accuracy of the proposed binBCO algorithm is given below. An UFLP with *i* candidate facility and *j* customers sites can be represented by a network with *i*+*j* nodes and *ij* arcs. In the UFLP, f_i is used to represent the cost of opening facility *i*, and c_{ij} is used to represent the cost of serving customer *j* from facility *i* or assigning customer *j* to facility *i*. It is assumed that $c_{ij} \geq 0$ for all *i*=1,2,…*I* and *j*=1,2,…*J* and $f_i \geq 0$ for all *i*=1,2,…*I* [18] .x_{ij} is the continuous variable representing the amount supplied to customer *j* from facility *i* and y_i is the 0-1 variable such that y_i= 1 if facility *i* is established and 0 otherwise [19]. The solution process of the UFLP is to find an optimal solution that satisfies all customer demand and minimizes the total cost (Eq. 23). The UFLP can be formally stated as [20]:

$$f(UFLP) = min\{\sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} + \sum_{i\in I} f_i y_i\} \tag{1}$$

Subject to:

$$\sum_{i\in I} x_{ij} = 1,\ j \in J \tag{2}$$

$$x_{ij} \leq y_i,\ i \in I\ and\ j \in J \tag{3}$$

[a] *Selcuk University, Faculty of Engineering, Department of Computer Engineering*
** Corresponding Author: Email: mskiran@selcuk.edu.tr;*

$$x_{ij} \in \{0,1\},\ i \in I \ and\ j \in J \tag{4}$$

$$y_i \in \{0,1\},\ i \in I \tag{5}$$

Constraint (2) makes sure that all demands have been met by a particular open site, and constraints (3)-(5) is to keep variables integer. With this model, we have to decide (*i*) the number of facility sites to be established and (*ii*) the quantities to be supplied from facility *i* to customer *j* such that the total cost (comprising fixed and variable costs) is minimized. When the locations of facilities which will be opened are determined, which customer will be served by a facility can be obtained easily. Therefore, we address a vector *A* of *n* variables where each variable in *A* is bit. Each bit (A_i, *i*=1,2,…,*n*) indicates that whether a facility is opened at location *i* or not. Due to the fact that the solution of each bee represents this vector, this vector plays a critical role in binBCO.

Location problems are some of the most widely studied problems in combinatorial optimization [21]. The UFLP which is one of the main problems in location problems is NP-hard even it has an uncapacitated assumption. This is perhaps the most common location problem, having been widely studied in the literature, both in theory and in practice [22]. Exact algorithms for UFLP do exist such as the dual approach of Erlenkotter [23], the primal-dual approaches of Körkel [24] and the branch-and-bound approach of Galvão and Raggi [25] but its NP-hard nature and high computation time makes heuristics and meta-heuristics the natural choice for larger instances. For this reason, some approximate methods have been proposed for solving UFLPs. These methods cannot guarantee to find the optimal solution but can obtain optimum or near optimum solution in a reasonable time. The meta-heuristics used to solve UFLPs are mainly genetic algorithm by Topçuoğlu et al. [26], tabu search by Sun [18], continuous and discrete particle swarm optimization algorithms by Sevkli and Güner [27] and Guner and Sevkli [28], simulated annealing by Yiğit et al. [20], artificial bee colony (ABC) algorithm by Kıran and Gündüz [29].

2.2. Bee Colony Optimization

The basic BCO was developed by inspiring natural behaviors of real bees between nest and food source. In the BCO, all bees try to construct a feasible solution for the optimization problems.

```
1.  Determine the number of bees in the hive. (NB)
2.  Assign the empty solutions to the artificial bees.
3.  Determine the stopping condition for the algorithm.(SC)
4.  While SC is met
        a.  Forward Pass: Allow the bees to construct self-partial
            tours.
        b.  Backward Pass: Return all bees to the hive and evaluate
            quality of partial solutions of bees. Test the loyalty of
            the bees to self-solutions and if any bee abandons self-
            solution, select a solution of recruiters (loyal bee to self
            -solutions) and assign this solution to the bee which
            abandons self-solution.
        c.  If the partial solutions of the bees are completed,
            evaluate the all solutions and determine the best one and
            clear the memory of the bees (delete the all solutions)
Report the best solution found by the bee population.
```

Fig.1. The basic steps of the BCO algorithm

At beginning of the search process, all bees are in the hive and the BCO consists of two phases sequentially realized. First phase is forward pass and in this phase, the bees create self-partial solution for the problem. The second phase is backward pass and in this phase, the bees that they are in the search space return to the hive in order to share their information about the partial solutions found. Being dependent on loyalty of the bees to the self-solutions, some of the bees leave self-partial solutions, and the partial solutions of the other bees are assigned for these bees. This process is named as recruitment process. After the partial tours are completed, an iteration of BCO is finished. The best solution in the population is determined and saved, and all solutions in the memory of bees are deleted in the basic version of BCO algorithm [4]. After the explanations are given above, the basic steps of the algorithm are presented in Fig. 1.

2.3. Proposed Method

In order to solve the UFLP, BCO algorithm is transformed to path improvement technique in the proposed method. The main difference between basic BCO and proposed approach is that the feasible solutions are assigned to the bees at the beginning of the algorithm. In the binBCO, the bees try to improve the self-solutions at the each iteration in the forward and backward passes instead of solution construction. After the bee population is created with feasible solutions for the UFLP, the best solution in the population is found. For each solution, objective function value (by Eq.1) and fitness of the solution by using Eq. 6 given below are calculated [30].

$$fit_i = \frac{1}{1+obj_i} \quad , i = 1,2,...,NB \tag{6}$$

where, fit_i is the fitness of the i^{th} solution (or bee), obj_i is the objective function value of i^{th} solution and *NB* is the number of bees.

In the forward pass of the binBCO, the new solutions are created for the bees to improve the solutions of the bees given as follows:

$$P = \frac{bestFit}{fit_i + bestFit},\ i = 1,2,...,NB \tag{7}$$

$$Pb_i = \frac{fit_i}{fit_i + bestFit} \tag{8}$$

$$B_{i,j} = \begin{cases} Best_j\,, & if\,(rnd_{i,j} < P) \\ B_{i,j}\,, & otherwise \end{cases} \tag{9}$$

where, *P* is the selection probability of the best solution, Pb_i is the selection probability of the bee, $bestFit$ is the fitness of the best solution in the population, $B_{i,j}$ is the j^{th} decision variable of the i^{th} solution, $Best_j$ is the j^{th} decision variable of best solution in the population and $rnd_{i,j}$ is a random number in range of [0,1]. In order to show improvement of the solution, an illustrative example is given below.

Assume that the best solution in the population is 011010, i^{th} solution (B_i) is 111001, the fitness of the best is 0.7 and the fitness of the B_i is 0.5. According to Eq.7, the selection probability of the best solution is calculated as 0.58, and the selection probability of the B_i solution is calculated 0.42 according to Eq.8. Assume that the random numbers generated for each dimension are 0.2, 0.9, 0.4, 0.3, 0.7, 0.6. According to these assumptions, the new solution is obtained as follows:

B_i

1	1	1	0	0	1

Best

0	1	1	0	1	0

Random Numbers

0.2	0.9	0.4	0.3	0.7	0.1

New Solution for B_i

0	1	1	0	0	0

After new solutions are obtained for each bee, the loyalty of the bees to the solutions is calculated as follows:

$$L_i = \frac{fit_i}{\sum_{i=1}^{NB} fit_i} \qquad (10)$$

where, L_i is the loyalty of i^{th} bee to self-solution and this is used for determining whether a bee abandon self-solution or not. If the loyalty of the bee is less than the mean loyalty of the population (obtained using Eq.11, where L_p is the mean loyalty value of the population), the bee abandons self-solution, and a random solution is generated for this bee.

$$L_p = \frac{1}{NB} \times \sum_{i=1}^{NB} fit_i \qquad (11)$$

where, L_p is the loyalty of the population and it is used to test the loyalty of the bees to self-solution. After proposed method is explained above, the algorithmic framework of the binBCO is presented in Fig.2.

```
1. Determine the number of bees in the hive. (NB)
2. Generate feasible solutions (binary values) for the bees.
3. Calculate objective function value for each bee.
4. Calculate fitness of the solutions of the bees by using Eq.6.
5. Determine the best solution in the population and save it as global best.
6. While a stopping condition is met
    a. For each bee
        i. Calculate selection probabilities of the best solution and the solution of bee by using Eq. 7 and 8.
        ii. For each decision variable
            1. If generated random number is less than the selection probability of best solution, the decision variable comes from the best solution, otherwise self-solution.
    b. Calculate objective function values and fitness of the solutions.
    c. Calculate the loyalty bees and population by using Eq.10 and Eq. 11.
    d. Fix the bees that they are not loyal to self-solution.
    e. Generate random solution for these bees.
    f. Determine the best solution in the population.
    g. If the best solution in population is better than the global best, replace them.
7. Report the best solution found by the population.
```

Fig.2. The binBCO algorithm

3. Computational Experiments

We examined the performance and accuracy of binBCO algorithm on the small and medium size test problems taken from OR-Library. The description for the problems is given in Table 1. Cap71-74 are the small size problems and they contain 16 decision variables. The rest of test suite is medium size problems and they contain 50 decision variables, and also the costs of the optimal solutions are presented in Table 1.

There are three control parameters in the binBCO algorithm. The first control parameter is the population size and it is analyzed by using different values such as 20, 40, 60, 80, 100. The second control parameter is the diversification ratio (DR). This parameter controls how many bees abandons self-solution at the each iteration. As previously mentioned, if the loyalty of a bee is less than the mean loyalty of population, this bee abandons self-solution, and a random solution is generated for this bee. In order to prevent much diversification in the population, DR parameter is set to 0.4 and it means that the random solutions are generated for 40% of bees that they abandoned self-solutions. If obtained number is floating number, it is rounded the nearest integer number. The last control parameter is the stopping condition for the binBCO algorithm. We used maximum iteration number for the stopping condition and it is set to 1000.

Table 1. The description of the test suite

Problem Name	Problem Size	Cost of Optimal Solution
Cap71	16 × 50	932615.75
Cap72	16 × 50	977799.40
Cap73	16 × 50	1010641.45
Cap74	16 × 50	1034976.98
Cap101	25 × 50	796648.44
Cap102	25 × 50	854704.20
Cap103	25 × 50	893782.11
Cap104	25 × 50	928941.75
Cap131	50 × 50	793439.56
Cap132	50 × 50	851495.33
Cap133	50 × 50	893076.71
Cap134	50 × 50	928941.75

Under these conditions, the binBCO algorithm is coded on Matlab platform, and the experiments are run on a PC with Intel i5 3.1 Ghz microprocessor and 4 Gb Ram. The proposed method is run 10 times for each problem and obtained results are presented in Table 2, 3 and 4. In the results tables (Table 1, 2 and 3), the best solution for each problem obtained by binBCO is written as bold font type. As seen from the result tables, while the population size is 100, better results is generally produced by the binBCO.

4. Results and Discussion

In this study, we proposed a binary version of the basic BCO algorithm, called as binBCO, and promising results for UFLPs were produced by the binBCO algorithm. While the number of decision variables or problem dimensionality is increased, the performance of the binBCO algorithm is decreased. In order to improve the local search ability of the binBCO algorithm, new local search techniques should be used. The other issue is to control diversification in the population. The random solutions are generated for the bees that they are not loyal self-solutions, this has caused more diversification in the population, and the intensification of the population has been weakened. In order to balance diversification in the population, we proposed DR control parameters for the bees that they are not loyal to self-solutions.

5. Conclusion and Future Works

The basic BCO algorithm was inspired by the intelligent behavior of real bees between nest and food sources. Increasing the popularity of BCO in solving the optimization problems shows that BCO algorithm is competitive method. In this study, a binary version of the basic BCO algorithm is developed and the solution construction approach is changed as solution improvement. The performance and accuracy of the new proposed method are tested on the UFLP which is a pure binary optimization problem. Experimental simulations show that proposed approach produces promising results. In our future works, we will include local search methods to the binBCO and compare the performance of binBCO with the other binary optimization methods.

Table 2. The results obtained under population sizes 20 and 40

Problem	Pop_Size=20				Pop_Size=40			
	Worst	Mean	Best	Std.Dev.	Worst	Mean	Best	Std.Dev.
Cap71	**932615.8**	**932615.8**	**932615.8**	**0**	935152.3	932869.4	932615.8	802.1
Cap72	981649.4	978292.1	977799.4	1227.2	982711.6	978290.6	977799.4	1553.4
Cap73	1014491.4	1011760.7	1010641.5	1323.8	**1012477**	**1011192.1**	**1010641.5**	**886.6**
Cap74	1037717.1	1035251	1034977	866.5	**1034977**	**1034977**	**1034977**	**0**
Cap101	801588.4	798688.6	796648.4	1538.9	800769.6	798229.1	796648.4	1241.7
Cap102	857956.1	856846	856004.4	669.6	859963.3	855686.6	854704.2	1612.7
Cap103	901977.1	895224	893782.1	2433.5	900022.1	895552.3	893782.1	2289
Cap104	935591.7	932481.3	928941.8	3062.6	942072.6	932111.5	928941.8	4563.7
Cap131	816940.8	807571.7	802160.2	4488.6	818222.8	806566.5	797306.9	6438.4
Cap132	887085.1	870204.9	861570.6	8548.9	878232.8	868713.4	854879	8269.9
Cap133	950367.4	918466.1	895642.5	16230.7	926371	910563.4	899963.4	8979
Cap134	983826.1	963197.5	945187.6	12522.8	965165.3	940782.6	934573.8	9367.2

Table 3. The results obtained under population sizes 60 and 80

Problem	Pop_Size=60				Pop_Size=80			
	Worst	Mean	Best	Std.Dev.	Worst	Mean	Best	Std.Dev.
Cap71	933568.9	932806.4	932615.8	401.9	**932615.8**	**932615.8**	**932615.8**	**0**
Cap72	981649.4	978569.4	977799.4	1623.3	982711.6	978290.6	977799.4	1553.4
Cap73	1014253.4	1011553.3	1010641.5	1284.1	1012477	1011559.2	1010641.5	967.4
Cap74	1037717.1	1036621	1034977	1415	1037717.1	1035525	1034977	1155.3
Cap101	800005	798245	796648.4	1201.9	800005	797826.2	796648.4	1176.9
Cap102	**858250.6**	**855629.6**	**854704.2**	**1334.2**	861025.5	856448.4	854704.2	1923.7
Cap103	898551.5	895440.6	894008.1	1897.9	898551.5	894915.5	893782.1	1447.9
Cap104	934587	930941.9	928941.8	2687.4	**934587**	**930635.3**	**928941.8**	**2726.9**
Cap131	809812	803524.7	797205.9	4721.1	812891.8	805886.4	799218.7	4934.4
Cap132	871487.2	863067.2	857054	4509.6	867959.6	858896.3	851495.3	5576.7
Cap133	909133.4	900254.7	894752	5110.4	**905245.8**	**899765.4**	**893733**	**4012.8**
Cap134	985002	956864.6	929477.6	21159.7	956707	938263.8	928941.8	8831.5

Table 4. The results obtained under population size 100

Problem	Pop_Size=100			
	Worst	Mean	Best	Std.Dev.
Cap71	**932615.8**	**932615.8**	**932615.8**	**0**
Cap72	**981649.4**	**978184.4**	**977799.4**	**1217.5**
Cap73	1012477	1011375.7	1010641.5	947.9
Cap74	**1034977**	**1034977**	**1034977**	**0**
Cap101	**799144.7**	**797803.2**	**796648.4**	**1168.6**
Cap102	861111.5	856276.3	854704.2	1913.5
Cap103	**898551.5**	**894859**	**893782.1**	**1411.4**
Cap104	934587	931764.4	928941.8	2975.3
Cap131	**804460.4**	**800422.8**	**794373.4**	**3250.5**
Cap132	**865076.8**	**858437.4**	**852762.9**	**3353.9**
Cap133	906857.7	900166.6	894095.8	3768.1
Cap134	**950386.9**	**937825.8**	**929477.6**	**6341**

References

[1] Dorigo M., Maniezzo V., Colorni A. Ant System: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics -Part B 26: 29–41, 1996.

[2] Dorigo M., Gambardella L. M., Ant Colony System: A cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation 1:53–66, 1997.

[3] Eberhart RC., Kennedy J, A new optimizer using particle swarm theory. In Proc. of Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 39-43, 1995.

[4] Teodorovic D., Bee colony optimization (BCO). Eds: Lim C.P., Jain L.C., Dehuri S., Innovations in swarm intelligence, Berlin, Heidelberg, Springer-Verlag, 39-60, 2009.

[5] Karaboga D,. An idea based on honey bee swarm for numerical optimization, Erciyes University, Technical Report-TR06, Kayseri/Turkey, 2005.

[6] Lucic P., Teodorovic D,. Bee System: modeling combinatorial optimization transportation engineering problems by swarm intelligence. In Preprints of the TRISTAN IV Triennial Symposium on Transportation Analysis, Sao Miguel, Azores Islands, Portugal, 441-445, 2001.

[7] Lucic P., Teodorovic D., Transportation Modeling: an artificial life approach. In Proc. of the 14th IEEE International Conference on Tools with Artificial Intelligence, Washington, 216-223, 2002.

[8] Lucic P., Teodorovic D., Computing with bees: attacking complex transportation engineering problems. International Journal of Artificial Intelligence Tools 12: 375-394, 2003.

[9] Lucic P., Teodorovic D., Vehicle routing problem with uncertain demand at nodes: the bee system and fuzzy logic approach. Ed.: Verdegay J.L., Fuzzy Sets in Optimization, Springer-Verlag, Heidelberg, Berlin, 67-82, 2003.

[10] Teodorovic D., Dell'Orco M., Bee colony optimization-a cooperative learning approach to complex transportation problems. In Proc. of the 10th Meeting Euro Working Group on Transportation, Poznan, Poland, 51-60, 2005.

[11] Teodorovic D., Dell'Orco M., Mitigating traffic congestion: solving the ride matching problem by bee colony optimization. Transportation Planning and Technology 31: 135-152, 2008.

[12] Teodorovic D., Lucic P., Markovic G., Dell'Orco M., Bee

colony optimization: Principles and Applications. Eds.: Reljin B., Stankovic S., In Proc. of the 8th Seminar on Neural Network Applications in Electrical Engineering – NEUREL 2006, Belgrade, 151-156, 2006.

[13] Markovic GZ., Teodorovic DB., Raspopovic VSA., Routing and wavelength assignment in all-optical networks based on the bee colony optimization. AI Communications 20: 273-285, 2007.

[14] Davidovic T., Selmic M., Teodorovic D., Scheduling Independent Tasks: Bee Colony Optimization Approach. In Proc. of the 17th Mediterranean Conference on Control&Automation, Thessaloniki, Greece, 1020-1025, 2009.

[15] Kıran MS., Bee Colony-based Driver-Line-Time Optimization. Master Thesis (In Turkish), 2010.

[16] Davidovic T., Ramljak D., Selmic M., Teodorovic D., Bee colony optimization for the p-center problem. Computers and Operations Research 38:1367-1376, 2011.

[17] Tedorovic D. Swarm intelligence systems for transportation engineering: Principles and applications. Transportation Research Part C: Emerging Technogies 16: 651-667, 2008.

[18] Sun M. Solving the uncapacitated facility location problem using tabu search. Computers & Operations Research 33: 2563-2589, 2006.

[19] Tcha D-W., Myung Y-S., Chung K-H. Parametric uncapacitated facility location. European Journal of Operational Research 86: 469-479, 1995.

[20] Yiğit V., Aydın ME., Türkbey O., Solving large-scale uncapacitated facility location problems with evolutionary simulated annealing. International Journal of Production Research 44: 4773-4791, 2006.

[21] Ghosh D., Neighborhood search heuristics for the uncapacitated facility location problem. European Journal of Operational Research 150: 150-162, 2003.

[22] Resende MGC., Werneck R.F., A hybrid multistart heuristic for the uncapacitated facility location problem. European Journal of Operational Research 174: 54-68, 2003.

[23] Erlenkotter D., A dual-based procedure for uncapacitated facility location. Operations Research 26: 992–1009, 1978.

[24] Körkel M., On the exact solution of large-scale simple plant location problems. European Journal of Operational Research 39: 157–73, 1989.

[25] Galvao RD., Raggi LA. A method for solving to optimality uncapacitated location problems. Annals of Operations Research 18: 225-244, 1989.

[26] Topçuoğlu H., Corut F., Ermiş M., Yılmaz G., Solving the uncapacitated hub location problem using genetic algorithms. Computers & Operations Research 32: 967-984, 2005.

[27] Sevkli M., Guner AR., A continuous particle swarm optimization algorithm for uncapacitated facility location problem. In Proc. of the Fifth International Workshop on Ant Colony Optimization and Swarm Intelligence - ANTS 2006, Brussels, Belgium, 316-323, 2006.

[28] Guner AR., Sevkli M, A discrete particle swarm optimization algorithm for uncapacitated facility location problem. Journal of Artificial Evolution and Applications pp. 1-9, 2008.

[29] Kıran MS., Gündüz M.. XOR-based artificial bee colony algorithm for binary optimization, Turkish Journal of Electrical Engineering and Computer Sciences, doi: 10.3906/elk-1203-104, 2013.

[30] Karaboga D., An idea based on honey bee swarm for numerical optimization, Erciyes University, Technical Report-TR06, Kayseri/Turkey, 2005.

International Journal of
Intelligent Systems and
Applications in Engineering

ISSN:2147-6799 www.atscience.org/IJISAE

Advanced Technology and Science

Original Research Paper

Preferences, Utility and Prescriptive Decision Control in Complex Systems

Yuri P. Pavlov*[a]

Received 28th June 2013, Accepted 18th September 2013

Abstract: *The evaluation of the preferences based utility function is a goal of the human cantered control (management) design. The achievement of this goal depends on the determination and on the presentation of the requirements, characteristics and preferences of the human behaviour in the appropriate environment (management, control or administration of complex processes). The decision making theory, the utility and the probability theory are a possible approach under consideration. This paper presents an approach to evaluation of human's preferences and their utilization in complex problems. The stochastic approximation is a possible resolution to the problem under consideration. The stochastic evaluation bases on mathematically formulated axiomatic principles and stochastic procedures. The uncertainty of the human preferences is eliminated as typically for the stochastic programming. The evaluation is preferences-oriented machine learning with restriction of the "certainty effect and probability distortion" of the utility assessment.The mathematical formulations presented here serve as basis of tools development. The utility and value evaluation leads to the development of preferences-based decision support in machine learning environments and iterative control design in complex problems.*

Keywords: *Preferences, Utility, Stochastic Approximation, Complex systems, Edgeworth box.*

1. Introduction

The aspiration for quantity measurements, estimations and prognosis at all phases of the decision making and problem-solving is natural. But this task is carried out with very scarce initial information, especially in the initial development phase in complex problems and situations. In the initial stage of a decision process the heuristic of the investigator is very important, because in most of the cases there is a lack of measurements or even clear scales under which to implement these measurements and computations. This stage is often outside of the strict logic and mathematics and is close to the art, in the widest sense of the word, to choose the right decision among great number of circumstances and often without associative examples of similar activity. The correct assessment of the degree of informativity and usability of these types of knowledge requires careful analysis of the terms measurement, formalization, and admissible mathematical operations under the respective scale, which do not distort the initial empirical information.

In the paper we describe approaches and methods for measurement and analytical presentations of empirical and scientific knowledge expressed as preferences. Due to multidisciplinary nature of the cognitive process and to multidisciplinary nature of the fields of applications our choice of scientific methods is oriented toward the utilization of the stochastic programming, the theory of measurement and utility theory [1], [7], [11], [13], [19]. In this manner we can pose the decision making problem as a problem of constructing value and utility functions based on stochastic recurrent procedures as machine learning, which can later be used in decision support, in intelligent information systems and human-adapted design process of optimization problems in complex systems with human participation. Validate mathematical evaluation of the human preferences as utility (value) is the first step in realization of a human-adapted design process and decision making [3], [11], [20].

The analytical description of the expert's preferences as value or utility function will allow mathematically the inclusion of the decision maker (DM) in the model description of the complex system "Technologist-process" [18]. Value based design is a systems engineering strategy which enables multidisciplinary design optimization. Value-driven design creates an environment that enables optimization by providing designers with an objective function [5]. The objective (value/utility) function inputs the important attributes of the system being designed, and outputs a score. In this way we introduce the Model-driven decision making. Model-driven decision making and control emphasizes access to and manipulation of a statistical, financial, optimization, or simulation models and uses data and parameters provided by users to assist decision process in analyzing a complex situation. The American psychologists Griffiths and Tenenbaum by analyzing intuitive evaluations in the conditions of repetitive life situations have proved the statistical optimality of human assessments [8]. The idea of this study is that humans process the new data about the surrounding world by interpreting them in the framework of a built in their consciousness probability model. That means that the Bayesian approach was a natural basis on which human beings form their decisions, using their previous empirical experience expressed as preferences [7], [11], [20]. In such case the utility theory and its prescription to make decision based on the optimal mathematical expectation of

[a] Institute of Information and Communication Technologies, Bulgarian Academy of Sciences
** Corresponding Author: Email: yupavlov14@hotmail.com;*

the utility has another scientific validation as methodology in the decision making.

We will demonstrate this system engineering, value driven approach within two examples, determination of the equilibrium points in competitive trading, modeled by the Edgworth box and the control design based on the human evaluation of the best growth rate of a biotechnological process.

2. Measurement, Scales, Preferences, Value and Utility Evaluation

The objective of value based decision making is to develop a mathematical framework (econometric) for management and modeling of complex systems. The aspiration for measurements, quantity estimations and prognosis is natural but the correct assessment requires careful analysis of the terms *measurement*, formalization and admissible mathematical operations. In complex processes, there is a lack of measurements or even clearly identifiable scales for the basic heuristic information. Internal human expectations and heuristic are generally expressed by qualitative preferences. The common sources of information in such a basic level are the human preferences. According to social-cognitive theories, people's strategies are guided both by internal expectations about their own capabilities of getting results, and by external feedback [3]. Probability theory and expected utility theory address decision making under these conditions [11].

The mathematical description on such a fundamental level requires basic mathematical terms like sets, relations and operations over them, and their gradual elaboration to more complex and specific terms like functions, operators on mathematically structured sets as well, and equivalency of these descriptions with respect to a given real object. In the last aspect of equivalency of the mathematical descriptions we enter the theory of *measurements and scaling* [13,19]**.**

People's preferences contain uncertainty of probabilistic nature due to the qualitative type of both the empirical expert information and human notions. A possible approach for solution of these problems is the *stochastic programming* [1, 15, 18]. The uncertainty of the subjective preferences could be considered as an additive noise that could be eliminated, as is typical in the stochastic approximation procedures. The main objective is the productive merger of the mathematical exactness with the empirical uncertainty in the human notions.

We start by a brief introduction in the measurement theory. *System with relations* (SR) is called the set $\boldsymbol{A}$ in conjunction with a set of relations $\boldsymbol{R_i}$ **,** $\boldsymbol{i}\in\boldsymbol{I}$, $\boldsymbol{I}=\{1,2,3,...,n\}$ defined over it and we denote it by $(\boldsymbol{A}, (\boldsymbol{R_i}), \boldsymbol{i}\in\boldsymbol{I})$. In this manner we introduce an algebraic structure in the set $\boldsymbol{A}$. Relation of *congruency* is called a relation of equivalency ($\approx$) (reflexive, symmetric and transitive relation) defined over the basic set $\boldsymbol{A}$, if the property of *substitution* is satisfied, i.e. from the fulfillment of relations $(x_1, x_2, x_3, ...,x_{hi})\in\boldsymbol{A}^{\mathrm{hi}}$ and $(x_j\approx y_j)$ for every j=1, 2, 3 4,...,$\mathrm{h_i}$ it follows that $\boldsymbol{R_i}(x_1, x_2, x_3, ...,x_{hi})=\boldsymbol{R_i}(y_1, y_2, y_3, ...,y_{hi})$ for $\forall$ $\boldsymbol{i}$, $\boldsymbol{i}\in\boldsymbol{I}$. We say that the relation of equivalency ($\approx_2$) is coarser than the equivalency ($\approx_1$), if the inclusion $(\approx_1)\subseteq(\approx_2)$ is satisfied. It is known that there always exists a coarsest relation ($\approx_A$) over the SR $(\boldsymbol{A}, (\boldsymbol{R_i}), \boldsymbol{i}\in\boldsymbol{I})$. This means that if two elements are in congruency $(x\approx_A y)$, then they are undistinguishable with respect to the properties in the set $\boldsymbol{A}$ (the real object under investigation), described with the set of relations $((\boldsymbol{R_i}), \boldsymbol{i}\in\boldsymbol{I})$. If we factorize the set $\boldsymbol{A}$ by the coarsest congruency ($\approx_A$), then in the factor set $\boldsymbol{A}/\approx_A$ the congruency ($\approx_A$) is in fact equality (=). A SR $(\boldsymbol{A}, (\boldsymbol{R_i}), \boldsymbol{i}\in\boldsymbol{I})$, in which the congruency ($\approx_A$) is coarsest is called *irreducible*. In this case SR $(\boldsymbol{A}/\approx_A,(\boldsymbol{R_i}), \boldsymbol{i}\in\boldsymbol{I})$ is irreducible.

A homomorphism is an image f, f: $\boldsymbol{A}\rightarrow\boldsymbol{B}$ between two systems with relations SR $(\boldsymbol{A},(\boldsymbol{R_i}), \boldsymbol{i}\in\boldsymbol{I})$ and SR $(\boldsymbol{B}, (\boldsymbol{S_i}), \boldsymbol{i}\in\boldsymbol{I})$ from the same type. The systems with relations SR $(\boldsymbol{A},(\boldsymbol{R_i}), \boldsymbol{i}\in\boldsymbol{I})$ and SR $(\boldsymbol{B}, (\boldsymbol{S_i}), \boldsymbol{i}\in\boldsymbol{I})$ are from the same type if for which $\forall$ $\boldsymbol{i}$, $\boldsymbol{i}\in\boldsymbol{I}$ and $(x_1, x_2, x_3, ...,x_{hi})\in\boldsymbol{R_i}$ is satisfied $\boldsymbol{R_i}(x_1, x_2, x_3, ...,x_{hi})\Leftrightarrow\boldsymbol{S_i}(f(x_1), f(x_2), f(x_3), ..., f(x_{hi}))$.

DEFINITION: We call k-dimensional scale every homomorphism from irreducible empirical system into a number system SR $(\boldsymbol{A}, (\boldsymbol{Q_i}), \boldsymbol{i}\in\boldsymbol{I})$.

The empirical system of relations SR $(\boldsymbol{A}, (\boldsymbol{R_i}), \boldsymbol{i}\in\boldsymbol{I})$ is an object from the reality with the properties described by the relations $((\boldsymbol{R_i}), \boldsymbol{i}\in\boldsymbol{I})$, while the numbered system of relations SR $(\boldsymbol{B}, (\boldsymbol{S_i}), \boldsymbol{i}\in\boldsymbol{I})$ is a mathematical object which reflects the properties of the real object. For example the set $\boldsymbol{B}$ could be the k-ary Cartesian product of the set of the real numbers $\boldsymbol{R}^{\kappa}$.

In the scale definition the correspondence f_0: $\boldsymbol{A}\rightarrow\boldsymbol{R}^{\kappa}$ is not simply defined. In general sense, there exists entire class of scales converting the irreducible empirical system of relations SR $(\boldsymbol{A}, (\boldsymbol{R_i}), \boldsymbol{i}\in\boldsymbol{I})$ into the number system SR $(\boldsymbol{R}^{\kappa}, (\boldsymbol{S_i}), \boldsymbol{i}\in\boldsymbol{I})$. We denote this class of homomorphisms by $\aleph(\boldsymbol{A}, \boldsymbol{R}^{\kappa})$. Every homomorphism of $\aleph(\boldsymbol{A}, \boldsymbol{R}^{\kappa})$ is injective because the empirical system is irreducible and surjective with regard to $f(\boldsymbol{A})$).

Let $\boldsymbol{A_0}$ be a subset of $\boldsymbol{A}$. We denote by $\mathrm{G}_A(\mathrm{A}_0)$ all injective inclusion (*partial endomorphism*) from SR $(\boldsymbol{A_0}, (\boldsymbol{R_i}), \boldsymbol{i}\in\boldsymbol{I})$ in SR $(\boldsymbol{A}, (\boldsymbol{R_i}), \boldsymbol{i}\in\boldsymbol{I})$. If a scale $f_0\in\aleph(\boldsymbol{A}, \boldsymbol{R}^{\kappa})$ is given, then we can characterize the whole class of scales $\aleph(\boldsymbol{A}, \boldsymbol{R}^{\kappa})$ in the following way: $\aleph(\boldsymbol{A}, \boldsymbol{R}^{\kappa}) =\{\gamma_{\circ} f_0 /$ where $\gamma\in\mathrm{G}_{\boldsymbol{R}^{\kappa}}(f_0(\boldsymbol{A}))\}$. In other words two scales are equivalent with precision up to a partial endomorphism $\gamma\in\mathrm{G}_{\boldsymbol{R}^{\kappa}}(f_0(\boldsymbol{A}))$. The elements of $\mathrm{G}_{\boldsymbol{R}^{\kappa}}(f_0(\boldsymbol{A}))$ are called *admissible manipulations* of the scale f_0 [19]. An example is the measurement of the temperature. If the scale $f_0(.)$ is the temperature in Celsius, then every partial endomorphism is an affine correspondence of the type $\gamma(\mathrm{x})=\boldsymbol{a}\mathrm{x}+\boldsymbol{b}$, $\boldsymbol{a}\in\boldsymbol{R}$, $\boldsymbol{b}\in\boldsymbol{R}$ and $\boldsymbol{a}>\boldsymbol{0}$. The temperature in Kelvin is determined by shifting the zero point by $\boldsymbol{b}$, $\boldsymbol{b}\in\boldsymbol{R}$ and by changing the magnitude by multiplying by $\boldsymbol{a}$, $\boldsymbol{a}>\boldsymbol{0}$.

From the definition of the measurement and scale it follows that there are infinitely many types of scales. In informal terms measurement is an operation in which a given state of the observed object is mapped to a given denotation. An example is the so-called *nominal scale* which is an expression of the equivalence of two phenomena only. Let $\mathbf{X}$ be the set of alternatives $(\mathbf{X}\subseteq\mathbf{R}^{\mathrm{m}})$. Let x and y be two alternatives $((x,y)\in\mathbf{X}^2)$. For this weakest scale the following axioms are valid:

1. $((\mathrm{x}\approx\mathrm{y} \vee \neg\mathrm{x}\approx\mathrm{y})\equiv 1) \wedge ((\mathrm{x}\approx\mathrm{y} \wedge \neg\mathrm{x}\approx\mathrm{y})\equiv 0) \wedge \mathrm{x}\approx\mathrm{x}$;
2. $(\mathrm{x}\approx\mathrm{y} \Rightarrow \mathrm{y}\approx\mathrm{x})$;
3. $((\mathrm{x}\approx\mathrm{y} \wedge \mathrm{x}\approx\mathrm{z}) \Rightarrow \mathrm{y}\approx\mathrm{z})$.

Here ($\approx$) denotes equivalence and $\neg(\approx)$ is the opposite (non-equivalence). The above three properties define the relation *equivalence*, which splits the set $\mathbf{X}$ into non overlapping subsets (classes of equivalence). In this scale only the Kronecker symbol may be used as a measure.

When the observed phenomenon allows to distinguish the differences between states and to compare them by preference a stronger scale needs to be used – the *ordering scale*. The preference in the ordering scale is denoted by $(x\succ y)$. In accordance with a long-standing tradition, $x\succ y$ is taken to represent "x is better than y". In this scale together with the above three axioms two more are satisfied:

4. $\neg(\mathrm{x} \succ \mathrm{x})$ for $\forall$ $\mathrm{x}\in\mathrm{X}$, $((\mathrm{x} \succ \mathrm{y}) \Rightarrow \neg(\mathrm{y} \succ \mathrm{x}))$;
5. $(\mathrm{x}\succ\mathrm{y} \wedge \mathrm{y}\succ\mathrm{z}) \Rightarrow \mathrm{x}\succ\mathrm{z}$.

If incomparable alternatives exist, then the scale is called *partial ordering*. Under these five axioms an analytical preferences representation by value function u(.) is searched for. A *value function* is a function u(.) for which it is fulfilled $((x, y) \in X^2, x \succ y) \Leftrightarrow (u(x) > u(y))$ [11]. In this definition, in addition to axioms (4, 5), weak connectedness is also assumed $\neg(x \approx y) \Rightarrow ((y \succ x) \vee (x \succ y))$. Depending on the type of the function – continuous, partially continuous or discrete – there exist different types of scales, measuring the above relations. A transformation with an arbitrary monotonous function leads to another ordinal scale (*admissible manipulations* γ, $\gamma \in G_{\boldsymbol{B}}$ $(f_0(\boldsymbol{A}))$. When using those scales, apart from comparison by magnitude, we can search the minimum and maximum of the function as feasible mathematical operations. Under this scale it is impossible to talk about distance between the different alternatives.

If together with the ordering of the alternatives, the distance between them can be evaluated, we can talk about *interval scale*. For these scales the distances between the alternatives have the meaning of real numbers. For these scales the central moments and the variance are sensible evaluations and have physical meaning, whereas the mathematical expectation depends on the origin of the scale and thus is unfeasible. The transition from one interval scale to another is achieved with affine transformation $\boldsymbol{x = ay + b, (x, y) \in X^2, a > 0, b \in R}$. Among these type of scales is also the measurement of the utility function by the so called "gambling approach". We emphasize that the calculations are done with numbers related to the distances between the alternatives, and not with the numbers relating to the alternatives themselves. For instance, if we say that a body is twice as warm as another in Celsius, this will not be true if the measurements were done in Kelvin.

A stronger scale is the *ratio scale*. This is an interval scale with fixed origin $\boldsymbol{x = ay, (x, y) \in X^2, a > 0}$. For example the weight measurement is in the ration scale. For these scales in addition to the previous 5 axioms the following additivity axioms are satisfied:

6. $(x = y \wedge z > 0) \Rightarrow ((x + z) > y)$;
7. $x+y=y+x$;
8. $(x=y \wedge z=q) \Rightarrow (x+z=y+q)$;
9. $q+(x+y)=(q+y)+x$.

The *absolute scale* is the most powerful. For it the zero and one are absolute and it is a one of a kind and unique scale.

2.1. Value Function and Measurement Scale

From practical point of view the empirical system of human preferences relations is a SR $(\boldsymbol{X}, (\approx), (\succ))$, where $(\approx)$ can be considered as the relation "indifferent or equivalent", and $(\succ)$ is the relation "prefer". We look for equivalency of the empirical system with the numbered system of relations SR (***R***-real numbers, (=), (>)). The *"indifference"* relation $(\approx)$ is based on $(\succ)$ and is defined by $((x \approx y) \Leftrightarrow \neg((x \succ y) \vee (x \prec y)))$. Let **X** be the set of alternatives $(\mathbf{X} \subseteq \mathbf{R}^m)$. A *Value function* is a function $(u^*: \mathbf{X} \rightarrow \mathbf{R})$ for which it is fulfilled [11]:

$((x, y) \in \mathbf{X}^2, x \succ y) \Leftrightarrow (u^*(x) > u^*(y))$.

It is proved that for a finite set of alternatives and partial ordering (axioms 4, and 5) there always exists such a function with precision up to monotonous transformation [7]. In this manner we can move from the language of binary relations and preferences to the language of control criteria as objective value function. The assumption of existence of a value function *u*(.) leads to the "*negatively transitive*" and "*asymmetric*" relation $(\succ)$, "*weak order*". A "*strong order*" is a "*weak order*" for which is fulfilled $(\neg(x \approx y) \Rightarrow ((x \succ y) \vee (x \prec y))$. The existence of a "*weak order*" $(\succ)$ over ***X*** leads to the existence of a "*strong order*" over $\boldsymbol{X}/\approx$ [7]. Consequently the assumption of existence of a value function u(.) leads to the existence of: asymmetry $((x \prec y) \Rightarrow (\neg(x \succ y))$, axiom 4), transitivity $((x \succ y) \wedge (y \succ z) \Rightarrow (x \succ z)$, axiom 5) and transitivity of the "indifference" relation $(\approx)$ (axiom 3).

The ordering scale was defined via homomorphisms, monotone functions. But if we are looking for the equivalency between SR $(\boldsymbol{X}, (\approx), (\succ))$ and SR (***R***-real numbers, (=), (>)) practically, the homomorphisms have to be not only monotonic but continuous as well. In this case the ordering in the real numbers ***R*** will be reflected in the empirical set ***X*** with the properties of the interval topology generated by the relation (>) in ***R***. Then the term for convergence in the measurements coincides with the standard generally accepted term for convergence [19].

2.2. Utility Function and Measurement Scale

According to the *Utility theory* let ***X*** be the set of alternatives and ***P*** is a set of probability distributions over ***X*** and $\boldsymbol{X} \subseteq \boldsymbol{P}$. A utility function *u*(.) will be any function for which the following is fulfilled:

$(p \succ q, (p,q) \in \mathbf{P}^2) \Leftrightarrow (\int u(.)dp > \int u(.)dq)$.

To every decision choice and action corresponds a probability distribution of appearance of final alternatives (results). The notation $(\succ)$ expresses the preferences of DM over ***P*** including those over ***X*** $(\boldsymbol{X} \subseteq \boldsymbol{P})$. The interpretation is that the integral of the utility function *u*(.) is a measure concerning the comparison of the probability distributions *p* and *q* defined over ***X*** (figure 1).

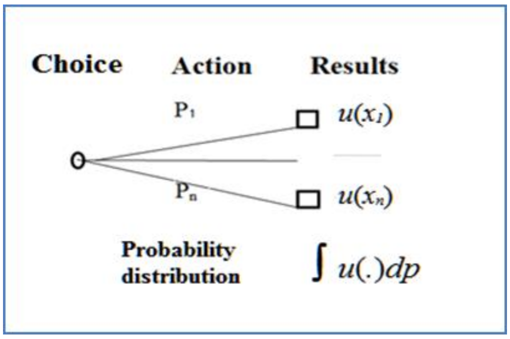


Fig.1. Probability distribution and utility function

There are different systems of mathematical axioms that give satisfactory conditions of a utility function existence. The most famous of them is the system of Von Neumann and Morgenstern's axioms [7]:

- (A.1) The *preferences* relations $(\succ)$ and $(\approx)$ are transitive, i.e. the binary preference relation $(\succ)$ is weak order**;**
- (A.2) *Archimedean Axiom*: for all $p,q,r \in \boldsymbol{P}$ such that $(p \succ q \succ r)$, there is an $\alpha, \beta \in (0,1)$ such that $((\alpha p + (1-\alpha)r) \succ q)$ and $(q \succ (\beta p + (1-\beta)r))$;
- (A.3) *Independence Axiom*: for all $p,q,r \in \boldsymbol{P}$ and any $\alpha \in (0, 1]$, then $(p \succ q)$ if and only if $((\alpha p + (1-\alpha)r) \succ (\alpha q + (1-\alpha)r))$.

Axioms (A1) and (A3) cannot give solution. Axioms (A1), (A2) and (A3) give solution in the interval scale (precision up to an affine transformation):

$((p \succ q) \Leftrightarrow (\int v(x)dp \succ \int v(x)dq) \Leftrightarrow (v(x) = \boldsymbol{a}u(x) + \boldsymbol{b}$,
$\boldsymbol{a}, \boldsymbol{b} \in \boldsymbol{R}, \boldsymbol{a} > 0, x \in \boldsymbol{X}))$.

It is known that the assumption of existence of a utility (value) function *u*(.) leads to the "*negatively transitive*" and "*asymmetric*" relation $(\succ)$ and to transitivity of the relation $(\approx)$. So far we are in the preference scale, the *ordering scale*. The assumption of equivalence with precision up to affine transformation has not been included. In other words we have only a value function. For value, however, the mathematical expectation is unfeasible, but we underline that the mathematical expectation is included in the definition of the utility function. For this reason it is accepted that (***X***⊆***P***) and that ***P*** is a convex set: $((q, p)\in \boldsymbol{P}^2 \Rightarrow (\alpha q+(1-\alpha)p)\in \boldsymbol{P}$, for $\forall\alpha\in[0,1])$.

Then utility $u(.)$ is determined in the interval scale [7]:

Proposition 1. If $((x\in X \wedge p(x)=1)\Rightarrow p\in \boldsymbol{P})$ and $(((q, p)\in \boldsymbol{P}^2) \Rightarrow ((\alpha p+(1-\alpha)q)\in \boldsymbol{P},\ \alpha\in[0,1]))$ are realized, then the utility function $u(.)$ is defined with precision up to an affine transformation: $(u_1(.)\approx u_2(.))\Leftrightarrow (u_1(.)=au_2(.)+b,\ a>0)$.

Following from this proposition, the measurement of the preferences is in the *interval scale*. That is to say, this is a utility function. Now it is obvious why in practice the gambling approach is used to construct the utility function in the sense of von Neumann. The reason is that to be in the interval scale the set of the discrete probability distributions ***P*** have to be convex. The same holds true in respect of the set ***X***. The utility function is evaluated by the "gambling approach". This approach consists within the comparisons between lotteries. A "lottery" is called every discrete probability distribution over X. We denote as <x, y,α> the simplest lottery: α is the probability of the appearance of the alternative *x* and (1-α) - the probability of the alternative *y*. In the practice, the utility measurement is based on the comparisons between lotteries as is shown in figure 2 [11].

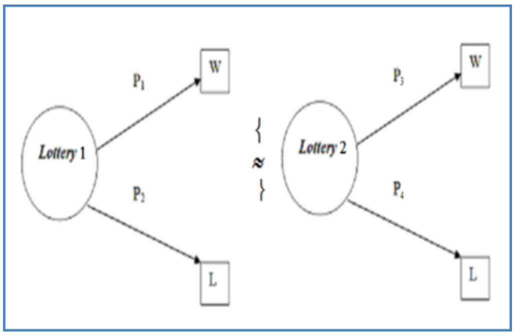


Fig.2. Gambling approach, comparisons of lotteries

The weak points of the gambling approach are the violations of the transitivity of the preferences and the so called "certainty effect" and "probability distortion" identified by the Nobel prizeman Kahneman and Tversky. The violations of the transitivity of the relation equivalence (≈) also lead to declinations in the utility assessment. All these difficulties explain the DM behavior observed in the Allais Paradox [2].

Following the research of Kahneman and Tversky and the debates about the well known Allais paradox, extensions and further developments of von Neumann's theory were sought [4], [10], [21]. Among these theories the rank dependent utility (RDU) and its derivative cumulative Prospect theory are currently the most popular. In the RDU the decision weight of an outcome is not just the probability associated with this outcome. It is a function of both the probability and the rank the alternative. Based on empirical researches several authors have argued that the probability weighting function has an inverse S-shaped form, which starts on concave and then becomes convex.

3. Utility And Value Stochastic Approximation Evaluation

Starting from the properties of the preference relation (≻) and indifference relation (≈), we propose the next stochastic approximation procedure for evaluation of the utility function $u(.)$. In correspondence with Proposition 1, it is assumed that (***X*** ⊆ ***P*)**, $((q,p)\in \boldsymbol{P}^2 \Rightarrow (\alpha q + (1-\alpha)p)\in \boldsymbol{P}$, for $\forall\alpha \in[0,1])$ and that utility function $u(.)$ exists. We define two sets:

$\boldsymbol{A}_{u^*}=\{(\alpha,x,y,z)/(\alpha u^*(x)+(1-\alpha)u^*(y))>u^*(z)\}$,

$\boldsymbol{B}_{u^*}=\{(\alpha,x,y,z)/(\alpha u^*(x)+(1-\alpha)u^*(y))>u^*(z)\}$,

where $u^*(.)$ is DM's empirical utility. The next proposition is in the foundation of the used stochastic approximation procedures [18]:

Proposition 2. We denote $\boldsymbol{A}_u=\{(\alpha,x,y,z)/(\alpha u(x)+(1-\alpha)u(y))>u(z)\}$. If $\boldsymbol{A}_{u1}=\boldsymbol{A}_{u2}$, then $u_1(.)=au_2(.)+b$, $a>0$**.**

The approximation of the utility function is constructed by recognition of the set ***A**u* [15], [18]. The proposed assessment is machine learning based on DM's preferences. The machine learning is a probabilistic pattern recognition ($\boldsymbol{A}_{u^*}\cap \boldsymbol{B}_{u^*}\neq\varnothing$) and the utility evaluation is a stochastic programming pattern recognition with noise (uncertainty) elimination**.** Key element in this solution is Proposition 2.

The evaluation procedure is presented as follows. The DM compares the "lottery" <x,y,α> with the simple alternative z, z∈**Z** **("better-≻, f(x,y,z,α)=(1)", "worse-≺, f(x,y,z,α)=(-1)", or "can't answer or equivalent- ~ , f(x,y,z,α)=0", f(.) denotes the qualitative DM answer).** This determines a learning point **((x,y,z,α), f(x,y,z,α)).** The following recurrent stochastic algorithm constructs the polynomial utility approximation:

$$u(x)=\sum_i c_i\Phi_i(x)\text{'}$$

$$c_i^{n+1}=c_i^{\,n}+\gamma_n\left[f(t^{n+1})-\overline{(c^n,\Psi(t^{n+1}))}\right]\Psi_i(t^{n+1})$$

$$\sum_{\mathbf{n}}\gamma_{\mathbf{n}}=+\infty,\ \sum_{\mathbf{n}}\gamma_{\mathbf{n}}^2<+\infty,\ \forall\mathbf{n},\gamma_{\mathbf{n}}>0\,.$$

In the formula are used the following notations (based on ***A*u**): ***t***$=(x,y,z,\alpha)$, $\psi_i(t)=\psi_i(x,y,z,\alpha) =\alpha\Phi_i(x)+(1-\alpha)\Phi_i(y)-\Phi_i(z)$, where $(\boldsymbol{\Phi}_i(x))$ is a family of polynomials. The line above the scalar product $v=(c^n,\Psi(t))$ means: $(\overline{v}=1)$, if $(v>1)$, $(\overline{v}=-1)$ if $(v<-1)$ and $(\overline{v}=v)$ if $(-1<v<1)$. The notation $(c^n,\Psi(t))=\alpha g^n(x)+(1-\alpha)g^n(y)-g^n(z)=G^n(x,y,z,\alpha)$ is a scalar product. The coefficients c_i^n take part in the polynomial presentation $g^n(x)=\sum_{i=1} c_i^n\Phi_i(x)$. The learning points are set with a pseudo random sequence. Practically the assessment process is the following. The expert (DM) relates intuitively the "learning point" (x,y,z,α)) to the set $\boldsymbol{A}_{u^*}$ with probability $D_1(x,y,z,\alpha)$ or to the set $\boldsymbol{B}_{u^*}$ with probability $D_2(x,y,z,\alpha)$. The probabilities $D_1(x,y,z,\alpha)$ and $D_2(x,y,z,\alpha)$ are mathematical expectation of $f(.)$ over $\boldsymbol{A}_{u^*}$ and $\boldsymbol{B}_{u^*}$ respectively,

$(D_1(x,y,z,\alpha)=M(f/x,y,z,\alpha))$ if $(M(f/x,y,z,\alpha)>0)$,

$(D_2(x,y,z,\alpha)=-M(f/\ x,y,z,\alpha))$ if $(M(f/x,y,z,\alpha)<0)$.

Let $D'(x,y,z,\alpha)$ is the random value:

$D'(x,y,z,\alpha)=D_1(x,y,z,\alpha)$ if $(M(f/x,y,z,\alpha)>0)$;

$D'(x,y,z,\alpha)=(-D_2(x,y,z,\alpha))$ if $(M(f/x,y,z,\alpha)<0)$;

$D'(x,y,z,\alpha)=0$ if $(M(f/x,y,z,\alpha)=0)$.

We approximate $D'(x,y,z,\alpha)$ by a function of the type $G(x,y,z,\alpha)=(\alpha g(x)+(1-\alpha)g(y)-g(z))$, where $g(x)=\sum_i c_i\Phi_i(x)$. The coefficients c_i^n take part in the polynomial approximation of $G(x,y,z,\alpha)$:

$$G^n(x,y,z,\alpha)=(c^n,\Psi(t))=\alpha g^n(x)+(1-\alpha)g^n(y)-g^n(z)$$

$$g^n(x)=\sum_{i=1}^{N} c_i^n\Phi_i(x)$$

The function $G^n(x,y,z,\alpha)$ is positive over $\boldsymbol{A}_{u^*}$ and negative over $\boldsymbol{B}_{u^*}$ depending on the degree of approximation of $D'(x,y,z,\alpha)$. The function $g^n(x)$ is the approximation of the utility function u(.). In another notation the stochastic procedure has the following form:

$$c_i^{n+1}=c_i^{\,n}+\gamma_n\left[D'(t^{n+1})+\xi^{n+1}-\overline{(c^n,\Psi(t^{n+1}))}\right]\Psi_i(t^{n+1})\ ,$$

$$\sum_{\mathbf{n}} \gamma_{\mathbf{n}} = +\infty, \sum_{\mathbf{n}} \gamma_{\mathbf{n}}^2 < +\infty, \forall \mathbf{n}, \gamma_{\mathbf{n}} > 0,$$

$$f(\boldsymbol{t}^{\mathbf{n}+1}) = \left[D'(\boldsymbol{t}^{\mathbf{n}+1}) + \xi^{\mathbf{n}+1} \right].$$

We follow the evaluation approach described in the well known books [11], [20]. DM compares the "lottery" <x,y,α>=(αx+(1−α)y) with the separate elements (alternative) z, z∈**X**. This lottery is of the simplest possible type and is sufficient for the utility evaluation. The expressed preferences, the answers of DM and comparisons are of cardinal (qualitative) nature and contain the inherent DM's uncertainty and errors. The stochastic convergence of the Potential function method (Kernel method) is analyzed in [1], [15], [18].

The same approach is used for of value evaluation. The difference is only within the form of the sets $\mathbf{A_{u*}}$ and $\mathbf{B_{u*}}$**.** Let $\mathbf{A_{u*}}$ and $\mathbf{B_{u*}}$ be the sets:

$A_{u*}=\{(x,y)\in R^{2m}/ (u*(x))>u*(y)\}$,

$B_{u*}=\{(x, y)\in R^{2m}/ (u*(x))<u*(y)\}$.

If there is a function F(x,y) of the form F(x,y)=f(x)-f(y), positive over A_{u*} and negative over B_{u*}, then the function f(x) is a value function, equivalent to the empirical value function u*(.). Such approach permits the use of stochastic "pattern recognition" for solving the problem. In the deterministic case it is true that $A_{u*}\cap B_{u*}=\emptyset$. In the probabilistic case it is true that $A_{u*}\cap B_{u*}\neq\emptyset$ and here have to be used the probabilistic pattern recognition [1, 12, 18].

4. Value Driven Decision Making and Equilibrium Analysis in Edgeworth Economic: Edgeworth Box and Competitive Trade

Competitive trade is a setting in which there are prices for two goods in question and many people who take these prices as given. Hence, the situation is as in the competitive market, except for the fact that we now consider two markets simultaneously. A useful tool for description the competitive trade is the Edgeworth Box. Essentially, it merges the indifference map between the parties in the trade by inverting one of the agents (individuals, consumers, markets and so on) diagram. Given two consumers $\boldsymbol{O}_1$ and $\boldsymbol{O}_2$, two goods, and no production, all non-wasteful allocations can be drawn in the box shown in figure 3.

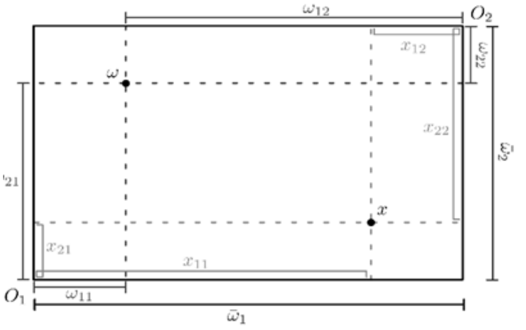
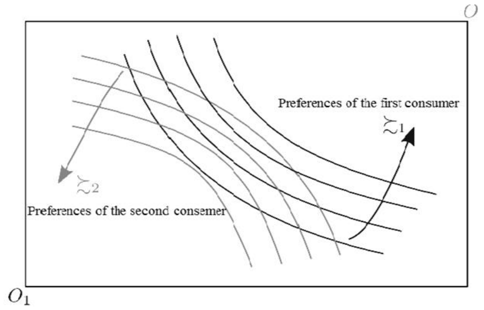


Fig.3. Edgeworth Box, initial endowment and allocations

Every point in the box represents a complete allocation of the two goods to the two consumers. Each of the two individuals maximizes his utility according to his preferences [11], [6]. The demand functions or the utility functions which represent consumers' preferences are convex and continuous, because in accordance with the equilibrium theory the preferences in are continuous, monotone and convex as is shown in figure 4 [6].



O
Preferences of the first consumer
$\succsim_1$
$\succsim_2$
Preferences of the second consemer
O_1

Fig.4. Convex indifference curves

Each consumer is characterized by an endowment vector, a consumption set, and regular and continuous preferences [6]. The two consumers are each endowed (born with) a certain quantity of goods. They have locally non-satiated preferences and initial endowments:

$(\boldsymbol{w}_1, \boldsymbol{w}_2) = ((w_{11}, w_{21}), (w_{12}, w_{22}))$.

In the box the vector $w = (w_1, w_2)$ is the total quantities of the two goods:

$$w_1 = w_{11} + w_{12}, w_2 = w_{21} + w_{22}.$$

An allocation $\boldsymbol{x}=(\boldsymbol{x}_1, \boldsymbol{x}_2) = ((x_{11}, x_{21}), (x_{12}, x_{22}))$ represents the amounts of each good that are allocated to each consumer. A no wasteful allocation $\boldsymbol{x}=(\boldsymbol{x}_1, \boldsymbol{x}_2)$ is one for which is fulfilled:

$$w_1 = x_{11} + x_{12}, w_2 = x_{21} + x_{22}.$$

In terms of aggregate amounts of the two agents, the total amounts needs to be equal to the total endowment of the two goods. The consumers take prices of the two goods $\boldsymbol{p} = (p_1, p_2)$ as given and maximize their utilities. The budget (income) set $B_i(\boldsymbol{p})$ of each consumer is given by: $B_i(\boldsymbol{p}) = \{\mathbf{x_i} \in \mathbf{R}_+^2 \ / \ \boldsymbol{px}_\mathbf{i} \le \boldsymbol{pw}_\mathbf{i}\}, (\mathbf{i}=1,2)$, where ($\boldsymbol{px}_i$) and ($\boldsymbol{pw}_i$) mean scalar products. For every level of prices, consumers will face a different budget set. The locus of preferred allocations for every level of prices is the consumer's offer curve.

An allocation is said to be Pareto efficient, or Pareto optimal, if there is no other feasible allocation in the Edgworth economy for which both are at least as well off and one is strictly better off. The locus of points that are Pareto optimal given preferences and endowments is the Pareto set, noted as ***P*** in figure 5. The part of the Pareto set in which both consumers do at least as well as their initial endowments is the Contract curve shown in figure 5 and noted as ***N*** (kernel of market game).

We are interested in the equilibrium point(s) of the process of exchange where is fulfilled the Walrasian equilibrium [6]. Walrasian equilibrium is a price vector ***p*** and an allocation ***x*** such that, for every consumer the prices (i.e. the terms of trade) are such that what one consumer (group of consumers) wants to buy is exactly equal to what the other consumer (group of consumers) wants to sell. In other words, consumers' demands are compatible with each other. We note the locus of points that are in Walrasian equilibrium as ***W*** (two points in figure 5).

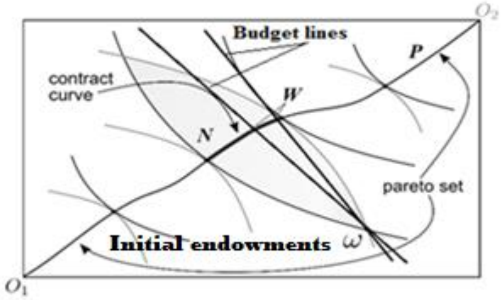


Fig.5. Pareto set and contract curve

In still other words, the quantity each consumer wants to buy at the given market prices is equal to what is available on the market. The following inclusion is true in the Edgworth economy [6]: $P \supset N \supset W$. In that sense a contract curve in the Edgeworth Box shows an exchange market in equilibrium and this is a particular representation of the Walrasian equilibrium theorem.

We had evaluated the consumer's preferences as value functions. In figure 6 are shown the indifference curves, calculations of the Pareto set ***P*** and the determination of the contract curve ***N***.

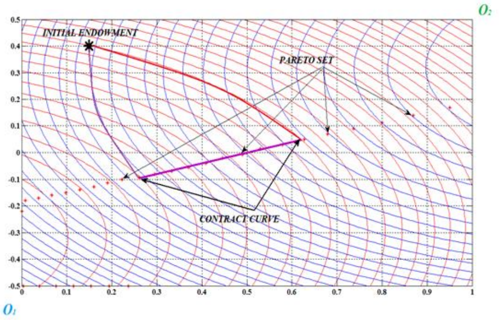


Fig.6. Real experiment-Pareto set and contract curve

The indifference curves in figure 6 are determined based on values functions evaluated by direct comparisons of couples of allocations $\mathbf{x}=(\mathbf{x}_1, \mathbf{x}_2) = ((x_{11}, x_{21}), (x_{12}, x_{22}))$. This is made through the discussed in the paper approach and algorithms for exact value function evaluation ($\mathbf{A}_{\mathbf{u}^*} \cap \mathbf{B}_{\mathbf{u}^*}=\emptyset$) [18]. After that we made quadratic approximation of the constructed value function. The little divergence from the theoretical convex requirements is due to the finite number of learning points and to the uncertainty in the expressed consumer's preferences. In the experiment for determination of the set $\mathbf{A}_{\mathbf{u}^*}$ and $\mathbf{B}_{\mathbf{u}^*}$we used a finite number of preferences expressed for couples of allocations ($\mathbf{x}=(\mathbf{x}_1, \mathbf{x}_2)$, $\mathbf{y}=(\mathbf{y}_1, \mathbf{y}_2)$):

$\mathbf{A}_{\mathbf{u}^*}=\{(x,y)\in\mathbf{R}^{2m}/ (\mathbf{u}^*(x))>\mathbf{u}^*(y)\}$,

$\mathbf{B}_{\mathbf{u}^*}=\{(x, y)\in\mathbf{R}^{2m}/ (\mathbf{u}^*(x))<\mathbf{u}^*(y)\}$.

The indifference curves could be determined by utility function evaluation also. The discussed previously in the paper stochastic procedures could be used for this purpose. In this case the learning points have to be defined as lotteries with Edgworth box allocations and consumers preferences in reference to learning triples of allocations. The described methodology and procedures allow for the design of individually oriented information systems [9]. Our experience is that the human estimation contains uncertainty at the rate of [10, 30] %. Such systems allow for exact evaluation of the Pareto set ***P***, a reasonable determination of the contract curve ***N*** and calculation of the Walrasian set ***W*** and may be autonomous or parts of larger decision support system [5, 6, 9]. The demands functions could be evaluated by direct comparisons or by the gambling approach. In that manner the incomplete information is compensated with the participation of qualitative human estimations.

In that manner we can state and solve the market-clearing equilibrium in principle and we can determine the contract curve and the Walrasian set in the Edgeworth box. The set of the Walrasian equilibriums ***W*** and the appropriate prices $\mathbf{p} = (p_1, p_2)$ are calculated based on the determined demand utility (value) functions and this is a meaningful prognosis of the market equilibrium. In that way can be forecast the competitive market equilibrium allocations $\mathbf{x}=(\mathbf{x}_1, \mathbf{x}_2) = ((x_{11}, x_{21}), (x_{12}, x_{22}))$ and the appropriate prices $\mathbf{p} = (p_1, p_2)$. The contract curves are specified on the individual consumers' preferences and show that there are possibilities to be made mutually advantageous trades. This means that one could unilaterally negotiate a better arrangement for everyone.

5. Utility Evaluation of The Best Growth Rate and Control Design

The complexity of the biotechnological systems and their singularities make them difficult objects for control. They are difficult to control because it is difficult to determine their optimal technological parameters [14], [17]. These parameters can depend on very complicated technological, ecological or economical market factors. Because of this in practice expert estimates are used. From outside the estimates are expressed by qualitative preferences of the technologist. The preferences themselves are in rank scale and bring the internal indetermination, the uncertainty of the qualitative expression. Our experience is that the human estimation of the process parameters of a cultivation process contains uncertainty at the rate of [10, 30] %. Because of this reason mathematical methods and models from the Utility theory and stochastic programming could be used in biotechnology. These stochastic methods, because of their essence, eliminate the uncertainty and could neutralize the wrong answers if one uses the gambling evaluation approach. Thus we achieve analytical math description of the complex system "Technologist-biotechnological process".

The approach used in the paper permits exact mathematical evaluation of the optimal specific growth rate of the fed-batch cultivation process according to the DM point of view. Let **Z** be the set of alternatives (**Z**={specific growth rates of the biotechnological process-μ}, **Z**≡[0, 0.6]) and **P** be a convex subset of discrete probability distributions over **Z**. The expert "preference" relation over **P** is expressed through ($\succ$) and this is also true for those over **Z** (**Z** $\subseteq$ **P**). The utility growth-rate function $U(.)$ is stochastically approximated by a polynomial [18].

$$U(\mu) = \sum_{i=1}^{n} c_i \mu^{\mathrm{i}}$$

This polynomial representation permits analytical determination of the derivative of the utility function and easy implementation in the optimal control theory [16]-[18]. Following the approach we are looking for pattern recognition of the sets of positive preferences $\boldsymbol{A}_{u^*}$ and negative preferences $\boldsymbol{B}_{u^*}$:

$\boldsymbol{A}_{u^*} = \{(x,y,z,\alpha)/ (\alpha u^*(x)+(1-\alpha)u^*(y))>u^*(z)\}$,

$\boldsymbol{B}_{u^*} = \{(x,y,z,\alpha)/ (\alpha u^*(x)+(1-\alpha)u^*(y))<u^*(z)\}$.

The star in the notations means an empirical estimate of the utility of the technologist. The utility function $U(\mu)$ itself is built as a recurrent procedure for the recognition of the set $\boldsymbol{A}_{u^*}$. The DM compares "lotteries" ($\alpha x+(1-\alpha)y$, $x,y\in\mathbf{Z}$, $\alpha\in[0,1]$) with simple alternatives $z\in\mathbf{Z}$ and the answer is determined from him (" better", "worse" or "indifference, equivalency or impossibility for explicit delimitation"). The Biotechnologist (DM) determines his answer (for every comparison): $f(x,y,z,\alpha)=1$ for ($\succ$), $f(x,y,z,\alpha)=-1$ for ($\prec$) and $f(x,y,z,\alpha)=0$ for ($\approx$). The function $f(x,y,z,\alpha)$ is a probability function, subjective characteristic of the DM depicturing intuition and empirical knowledge and also including subjective and probability uncertainty of the answers. In the recurrent procedure "the training point" ($x,y,z,\alpha, f(x,y,z,\alpha)$) is treated as point from the set $\boldsymbol{A}_u$ with probability $D_1(x,y,z,\alpha)$ or a point from $\boldsymbol{B}_u$ with probability $D_2(x,y,z,\alpha)$). We suppose that (x,y,z,α) are given by probability distribution $F(x,y,z,\alpha)$. In fact this is a pseudo-random Lp_τ sequence of Sobol. Then probabilities $D_1(x,y,z,\alpha)$ and $D_2(x,y,z,\alpha)$ are the conditional mathematical expectations of $f(.)$ over the sets $\boldsymbol{A}_u$ and $\boldsymbol{B}_u$, respectively. With $D'(x,y,z,\alpha)$ we denote the conditional random value:

$$D'(x,y,z,\alpha) = \begin{cases} D_1(x,y,z,\alpha), & \text{when } M(f/x,y,z,\alpha)>0, \\ -D_2(x,y,z,\alpha), & \text{when } M(f/x,y,z,\alpha)<0, \\ 0, & \text{when } M(f/x,y,z,\alpha)=0. \end{cases}$$

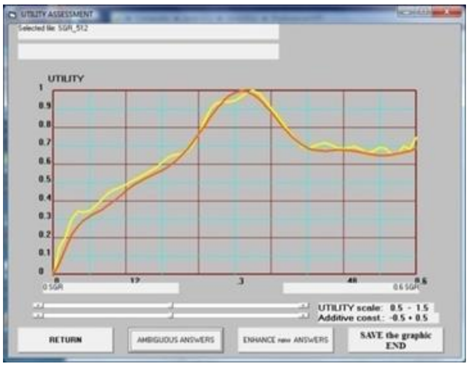
The measurable function $D'(x,y,z,\alpha)$ is approximated by function of the type $G(x,y,z,\alpha)=(\alpha g(x)+(1-\alpha)g(y)-g(z))$. The function $g(x)$ is an approximation of the utility $U(.)$. The coefficients c_i^n take part in:

$$g^n(x)=\sum_{i=1}^{N} c_i^n \Phi_i(x)$$

$$(c^n,\Psi(t))=\alpha g^n(x)+(1-\alpha)g^n(y)-g^n(z)=G^n(x,y,z,\alpha)$$

The function $G^n(x,y,z,\alpha)$ is positive over $\boldsymbol{A}_{\mathbf{u}}$ and negative over $\boldsymbol{B}_{\mathbf{u}}$ depending on the degree of approximation of $D'(x,y,z,\alpha)$. In fact total recognition is impossible, because of the wrong preferences of the technologist caused by the uncertainty within his preferences ($\boldsymbol{A}_{u^*}\cap\boldsymbol{B}_{u^*}\neq\varnothing$). The process of the recognition of the sets $\boldsymbol{A}_{u^*}$ и $\boldsymbol{B}_{u^*}$ is shown on the figure (7). The polynomial approximation of the DM utility $U(\mu)$ is the smooth line in figure (7). The maximum of the utility function determines the „best" growth rate of the fed-batch process after the technologist. A session with 128 questions learning points $(x,y,z,\alpha, f(x,y,z,\alpha))$ takes no more than 45 minutes.

The Value based control design is determined by the solution of the next optimal control problem: ***max***$(U(\mu))$, where the variable μ is the specific growth rate, ($\mu\in[0, \mu_{\max}]$, $D\in[0, D_{\max}]$).


UTILITY ASSESSMENT

Selected file: SGR_512

UTILITY scale: 0.5 - 1.5

Additive const: -0.5 + 0.5

Fig.7. Growth rate utility

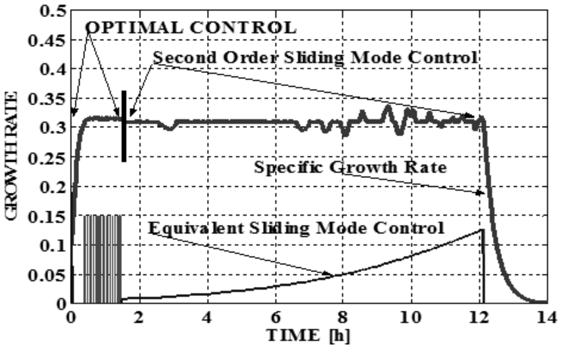


Fig.8. Stabilization of the fed-batch process

Here $U(\mu)$ is an aggregation objective function (the utility function) and D is the control input (the dilution rate):

$$\mathbf{max}(U(\mu)),\ \mu\in[0,\mu_{\max}],\ t\in[0,T_{\text{int}}],\ D\in[0,D_{\max}]$$

$$\dot{X}=\mu X-DX$$

$$\dot{S}=-k\mu X+(So-S)D$$

$$\dot{\mu}=m(\mu_m\frac{S}{(K_S+S)}-\mu)$$

The differential equation describes a *continuous biotechnological process*. The Monod-Wang model permits exact linearization to Brunovsky normal form following the procedures in papers [16], [17]. The optimal solution is determined with the use of the equivalent Brunovsky normal form of the differential equation above:

$$\dot{Y}_1=Y_2$$

$$\dot{Y}_2=Y_3$$

$$\dot{Y}_3=W.$$

In the formula, W denotes the control input of the Brunovsky model. The two differential equations above are equivalent as objects for control. The vector (Y_1, Y_2, Y_3) is the new state vector:

$$Y_1=u_1$$

$$Y_2=u_3(u_1-ku_1^2)$$

$$Y_3=u_3^2(u_1-3ku_1^2+2k^2u_1^3)+m(\mu_m\frac{u_2}{(K_S+u_2)}-u_3)(u_1-ku_1^2)$$

$$\begin{pmatrix}u_1\\u_2\\u_3\end{pmatrix}=\Phi(X,S,\mu)=\begin{pmatrix}\dfrac{X}{S_0-S}\\ S\\ \mu\end{pmatrix}$$

The derivative of the function Y_3 determines the interconnection between W and D, the inputs of the equivalent models. The control design is a design based on the Brunovsky normal form and the application of the Pontrjagin's maximum principle step by step for sufficiently small time periods T. The optimal control law has the analytical form [16], [17]:

$$D_{opt}=\mathbf{sign}\left(\left(\sum_{i=1}^{6}ic_i\mu^{(i-1)}\right)(T-t)\left[\frac{(T-t)\mu(1-2kY_1)}{2}-1\right]\right)D_{\max},$$

$$where:\quad \mathbf{sign}\,(r)=1, r>0, \mathbf{sign}\,(r)=0, r\le 0.$$

The sum is the derivative of the utility function. It is clear that the optimal "time-minimization" control is determined from the sign of the utility function derivative. The control input is $D=D_{\max}$ or D = 0. The solution is in fact a ***"time-minimization"*** control (if the time period T_{int} is sufficiently small). The control brings the system back to the set point for minimal time in any case of specific growth rate deviations.

The control law *of the fed-batch process* has the same form because $D(t)$ is replaced with $F(t)/V(t)$ in Monod-Wang model [16], [17]:

$$\dot{X}=\mu X-\frac{F}{V}X,$$

$$\dot{S}=-k\mu X+(So-S)\frac{F}{V},$$

$$\dot{\mu}=m(\mu_m\frac{S}{(K_S+S)}-\mu),$$

$$\dot{V}=F,$$

$$\dot{E}=k_2\mu E-\frac{F}{V}E$$

Thus, the feeding rate $F(t)$ takes $F(t)=F_{\max}$ or $F(t)=0$, depending on $D(t)$ which takes $D=D_{\max}$ or $D=0$. We conclude that the control law brings the system to the set point (optimal growth rate) with time minimization control, starting from any deviation of the specific growth rate as is shown in figure 8. We use this control law as a main part in a more complex chattering control law for stabilization of the system in the "best" growth rate [14], [16]-[18]. The deviation of the fed-batch process with this chattering control is shown on figure (8). After the stabilization of the system in equivalent sliding mode control position the system can be maintained around the optimal parameters with sliding mode control.

6. Conclusions

Human values (utilities) are integral part of the decision making process of the individual. They are the internal motivation for determining the main objective in the goal-oriented systems. Unfortunately, in most scientific investigations and developments, the subjective values and probabilistic expectations are not explicitly related and directly oriented towards the considered problem. In this aspect, especially important is the task of connecting the two contradicting tendencies: the requirement of ordinal information from mathematical and computational point of view and the cardinal nature of the empirical knowledge.

One of the possible scientific approaches in regards to these problems is that of multiattribute utility. In this manner in difficult for formalization and even verbally expressed weakly structurized problems and complex events we introduce the strict analytical approach, as analysis and analytically based synthesis, which allows for logically sound and mathematically precise decision formation. We achieve analytic model description of complex process with human participation. Such models ensure exact mathematical descriptions of problems in various areas for which the quantitative modeling is difficult: economics, biotechnology, ecology, and so on. These models guarantee that the powerful optimal control theory could be applied for exact mathematical solutions in such complex areas.

By the Edgworth box and the growth rate control examples we saw that the utility approach permits exact mathematical evaluation according to the consumers' point of view even though the human thinking is qualitative and pierced by uncertainty. Measurement, Expected utility theory and stochastic programming are some of the approaches for attainment of these purposes. These examples show that the presented methodology and mathematical procedures allow for the design of individually oriented decision support systems. Such systems may be autonomous or parts of larger intelligent information or decision support systems and can permit reasonable optimal solutions and prognoses.

References

[1] M. A. Aizerman, E. Braverman and L. Rozonoer, Potential function method in the theory of machine learning, Moscow, Russia, Nauka (In Russian), 1970.

[2] M. Allais, Le comportement de l'homme rationnel devant le risque: Critique des postulats et axiomes de l'école américaine, Econometrica, 21, 503-546, 1953.

[3] A. Bandura, Social foundations of thought and action: A social cognitive theory, Englewood Cliffs, NJ: Prentice-Hall,1986.

[4] M. Cohen, J.-Y. Jaffray, Certainty effect versus probability distortion: An experimental analysis of decision making under risk, Journal of Experimental Psychology: Human Perception and Performance, 14(4), 554-560, 1988.

[5] P. Collopy, P. Hollingsworth, Value-driven design. AIAA Paper 2009-7099, American Institute of Aeronautics and Astronautics, Reston, VA, 2009.

[6] I. Ekeland, Elements d'économie mathématique, Hermann, 1979, Russian translation: Mir, 1983.

[7] P. Fishburn, Utility theory for decision-making, New York, Wiley, 1970.

[8] T. L. Griffiths, J. B. Tenenbaum, Optimal prediction in everyday cognition, Psychological Science, 17 (9), 767-773, 2006.

[9] J. Hahn, H. Kuhn, Designing decision support systems for value-based management: A survey and an architecture, Decision Support Systems-DSS, Vol. 53, Issue 3, 591-598, 2012.

[10] D. Kahneman, A. Tversky, Prospect theory: An analysis of decision under risk, Econometrica, 47, 263-291, 1979.

[11] R. Keeney, H. Raiffa, Decision with multiple objectives: Preferences and value trade-offs, Cambridge & New York: Cambridge University Press, 1993.

[12] J. Kivinen, A. Smola and R. Williamson, Online learning with kernels, IEEE Transactions on Signal Processing, 52(8), 2165-2176, 2004.

[13] D. H. Krantz, R. D. Luce, P. Suppes and A. Tversky, Foundations of measurement, Vol. I., II, III. New York, Academic. Press, (1971, 1989, 1990).

[14] R. Neeleman, Biomass performance: Monitoring and control in biopharmaceutical production, Thesis, Wageningen University, Belgium, 2002.

[15] Y. Pavlov, Subjective Preferences, Values and Decisions: Stochastic Approximation Approach, Proceedings of Bulgarian Academy of Sciences (Comptes rendus de L'Academie bulgare des Sciences), Tome 58, N4, 367-372, 2005. Available: http://cat.inist.fr/?aModele=afficheN&cpsidt=16768115

[16] Y. Pavlov, Equivalent forms of Wang-Yerusalimsky kinetic model and optimal growth rate control of fed-batch cultivation processes, Online Journal Bioautomation, 11, 1-13, 2008. Available: http://core.kmi.open.ac.uk/display/791332

[17] Y. Pavlov, Specific growth rate and sliding mode stabilization of fed-batch processes, Global Journal of Computer Science and Technology, 11(20), (1.0), 19-29, 2011.Available: http://computerresearch.org/stpr/index.php/gjcst/article/viewArticle/935

[18] Y. Pavlov and R. Andreev, Decision control, management, and support in adaptive and complex systems: Quantitative models, Hershey, Pennsylvania, PA: IGI Global, 2013.

[19] J. Pfanzagl, Theory of Measurement, Physical-Verlag, Wurzburg-Wien, 1971.

[20] H. Raiffa, Decision Analysis, New York: Addison-Wesley Reading Mass, 1968.

[21] D. Shmeidler, Subjective probability and expected utility without additivity, Econometrica, 57(3), 571-587, 1989.

International Journal of
Intelligent Systems and
Applications in Engineering

ISSN:2147-6799 www.atscience.org/IJISAE

Advanced Technology and Science

Original Research Paper

Statistical Methods for Quantitatively Detecting Fungal Disease from Fruits' Images

Jagadeesh D.Pujari[1], **Rajesh Yakkundimath***[2], **Abdulmunaf S.Byadgi**[3]

Received 10th September 2013, Accepted 12th December 2013

Abstract: In this paper, we have proposed statistical methods for detecting and classifying fungal disease. The classification is done based on disease severity levels. In this work, we have considered fungal disease symptoms affected on fruits like mango, pomegranate and grape. In this study, images of fruits affected by different fungal symptoms are collected and categorized based on disease severity as partially affected, moderately affected, severely affected and normal. Statistical features using block wise, Gray Level Co-occurrence Matrix (GLCM), Gray Level Run-length Matrix (GLRM) are extracted from these images. The Nearest Neighbor classifier using Euclidean distance is used to classify images as partially affected, moderately affected, severely affected and normal. The average classification accuracies are 91.37% and 86.715% using GLCM and GLRM features. The average classification accuracy has increased to 94.085% using block wise features.

Keywords: *Fungal disease, Disease severity, Fruits, Statistical features, Euclidean distance, Nearest Neighbor.*

1. Introduction

Fruit industry is a major industry which contributes 20% of the nation's growth. Increase in the production and productivity is largely due to the adoption of improved technologies, which include quality planting material, balanced nutrients and timely protection against major insect-pests and diseases. India is the second largest producer of fruits with a production of 44.04 million tonnes from an area of 3.72 million hectares. This accounts 10% of the world fruit production. A large variety of fruits are grown in India of which apple, citrus, banana, grape, mango, guava, are the major ones. Also, India is a large low cost producer of fruit, and horticulture has huge export potential.

In spite of the fact that India is blessed with a wide range of soil and climatic conditions for growing large number of horticultural crops, there are still several constraints which adversely affect development of a sound horticulture industry. Due to improper cultivation of fruits, lack of maintenance and manual inspection there has been a decrease in production of good quality of fruits. Farmers are finding difficulty, especially in finding the fruits affected by diseases which results in huge loss of revenue to the farmers and the nation. Non adoption of adequate and timely control measures against pests and diseases also cause major fruit losses. In the absence of comprehensive knowledge, disputes over costs, benefits, and the potential for harm of chemical pesticides easily become polarized [31]. Farmers are also concerned about the huge costs involved in these activities and severe loss. The cost intensity, automatic correct identification of diseases based on their particular symptoms is very useful to farmers and also agriculture scientists. Detection of diseases is a major challenge in horticulture / agriculture science. Development of proper methodology, certainly of use in these areas. One of the main concerns of scientists is the automatic disease diagnosis and control [15].

Computer vision systems developed for agricultural applications, namely detection of weeds, sorting of fruits in fruit processing, classification of grains, recognition of food products in food processing, medicinal plant recognition etc. In all these techniques, digital images are acquired in a given domain using digital camera and image processing techniques are applied on these images to extract useful features that are necessary for further analysis. To know the state-of-the-art in automation of the task/activities in horticulture field and automatic detection of fruit disease using computer vision techniques, a survey is made. The gist of a survey which carried out is given as follows.

(Jagadeesh D.Pujari et al; 2013) proposed grading and classification of anthracnose fungal disease in mangoes. Different types of segmentation techniques were used to separate and grade percentage of affected areas. GLRM was used to extract texture features and further classified fungal affected mango images from normal using Artificial Neural Network (ANN) classifier. (Sudheer reddy bandi et al; 2013) proposed machine vision and image processing techniques in sleuthing the disease mark in citrus leaves. Citrus leaves were investigated using texture analysis based on the Color Co-occurrence Matrix (CCM) and classified using various classifiers. (Shiv Ram Dubey et al; 2012) proposed image processing based approach to evaluate diseases of apple. Local binary features were extracted from the segmented image, and finally images were classified using a multi-class Support Vector Machine (SVM). (Patil et al; 2012) describes the method for extraction of color & texture features of diseased leaves of maize. The textures features like correlation, energy, inertia and homogeneity were obtained by computing GLCM. (Jayamala K. Patil and Raj Kumar, 2011) have provided advances in various methods used to study plant diseases/traits using image processing. The methods studied were for increasing throughput and reducing subjectiveness arising from human experts in detecting the plant diseases. (D. Moshou et al; 2011) developed a prototype system

[1] *S.D.M.College of Engg. &Tech, Dharwar – 580 008, India*
[2] *KLE.Institute of Technology, Hubli – 580 030, India*
[3] *University of Agricultural Sciences, Dharwar – 580005, India*
* *Corresponding Author: Email: rajeshymath@gmail.com*

for detection of plant diseases in arable crops automatically at an early stage of fungal disease development and during field operations. Hyperspectral reflectance and multi-spectral imaging techniques were developed for simultaneous acquisition of images. An intelligent multi-sensor fusion decision system based on neural networks was developed to predict the presence of diseases. A robust multi-sensor platform integrating optical sensing, Geostationary Positioning System (GPS) and a data processing unit was constructed and calibrated. (D.S.Guru et al., 2011) have presented a novel algorithm for extracting lesion area and application of neural network to classify tobacco seedling diseases. First order statistical texture features were extracted from lesion area and Probabilistic Neural Network (PNN) is employed to classify anthracnose and frog-eye spots present on tobacco seedling leaves. (H. Al-Hiary et al., 2011) have evaluated a software solution for automatic detection and classification of plant leaf diseases. The affected area was segmented and texture analysis was done using CCM. Neural network classifier was used to classify various plant diseases. (Di Cui et al; 2010) reports research outcomes from developing image processing methods for quantitatively detecting soybean rust severity from multi-spectral images. To achieve automatic rust detection, an alternative method of analyzing the centroid of leaf color distribution in the polar coordinate system was investigated. Leaf images with various levels of rust severity were collected and analyzed. (Qing Yao et al., 2009) presented an application of image processing techniques and SVM for detecting rice diseases using shape and texture features. (Dae Gwan Kim et al; 2009) investigated the potential of using color texture features for detecting citrus peel diseases. Classification models were constructed using the reduced texture feature sets through a discriminant function based on a measure of the generalized squared distance. (Geng Ying et al., 2008) have provided various methods of image preprocessing techniques for recognition of crop diseases. (Di Cui et al; 2008) proposed a method to detect the infection and severity of of soybean rust. The test performed using multispectral image sensor could quantitatively detect soybean rust compared to laboratory-scale research. (Kuo-Yi Huang, 2007) have presented an application of neural network and image processing techniques for detecting and classifying phalaenopsis seedling diseases. The texture features using GLCM and color features were used in the classification procedure. A Back Propagation Neural Network (BPNN) classifier was employed to classify phalaenopsis seedlings diseases. (Alexander A. doudkin et al., 2007) have proposed a neural network clusterization algorithm for segmentation of the color images of crop field infected by diseases that change usual color of agricultural plants. (Pydipati et al., 2006) have used a computer vision and image processing techniques in the early detection and classification of diseased citrus leaves from normal citrus leaves. The color texture features using CCM was used as input to various classifiers. (Hamid Muhammed and Hammed, 2005) work was concerned with characterizing and estimating fungal disease severity in a spring wheat crop. This goal can be accomplished by using a reference data set consisting of hyperspectral crop reflectance data vectors and the corresponding disease severity field assessments. (Marc Lefebvre et al., 1993) have presented the problem in automatzing pulp sampling of potatoes such as their shape, color or texture in order to detect viral diseases. The sprouts, where the viral activity is maximum, were then detected by an active vision process operating on multiple views.

Most of the published work has mainly focused on generic diseases affected on single crop/fruit type. Most fruits diseases are caused by bacteria, fungi, virus, etc of which fungi are responsible for a large number of diseases in fruits. Fruits get affected are common and not much work is cited on fruits like mango, pomegranate and grape affected by fungal disease. Although several image processing approaches have been presented, no attempts are made for quantitatively detecting fungal affected fruits and classifying based on disease severity levels. In this paper, we have developed a methodology for recognition of fungal disease severity and determine whether fruit is partially affected, moderately affected, severely affected or normal. The samples of fungal affected images are shown in (Figure.1 & 2).

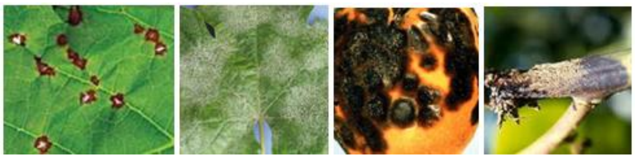


Figure 1. Images showing the visual symptoms cause by fungal disease

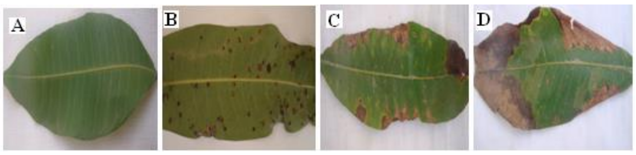


Figure 2. Images corresponding to different fungal disease severity levels: normal (A); partially affected (B); moderately affected(C); severely affected (D).

The paper is organized into four sections. Section.2 gives proposed methodology. Section.3 describes results and discussion. Section.4 gives conclusion of the work.

2. Proposed Methodology

In the present work, tasks like image acquisition, preprocessing, feature extraction, classification are carried out. The classification tree is given in (Figure.3). The detailed block diagram of adopted methodology is shown in (Figure.4).

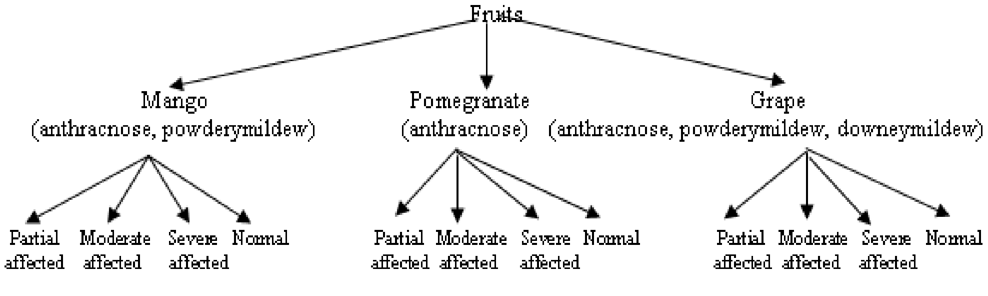


Figure 3. Classification tree

Table 1. Scientific classification of fungal symptoms affected on each fruit type

Fungal Symptom	*Causal organism*	*Family*	*Order*	*Class*	*Subdivision*	*Affected part*
Mango Anthracnose	Glomerella cingulata	Glomerellaceae	Incertaesedis	Sordariomycetes	Sordariomycetidae	stem, leaf, fruit
Mango Powdery mildew	Oidium mangiferae	Erysiphaceae	Erysiphales	Leotiomycetes	Leotiomycetidae	stem, leaf, fruit
Pomegranate Anthracnose	Glomerella cingulata	Glomerellaceae	Melanconiales	Sordariomycetes	Pezizomycotina	stem, leaf, fruit
Grape Anthracnose	Elsinoë ampelina	Elsinoaceae	Incertaesedis	Dothideomycetes	Dothideomycetidae	stem, leaf, fruit
Grape Downey mildew	Plasmopara viticola	Peronosporaceae	Pleosporales	Oomycota	Mastigomycotina	stem, leaf, fruit
Grape powdery mildew	Uncinula necator	Erysiphaceae	Erysiphales	Leotiomycetes	Ascomycota	stem, leaf, fruit

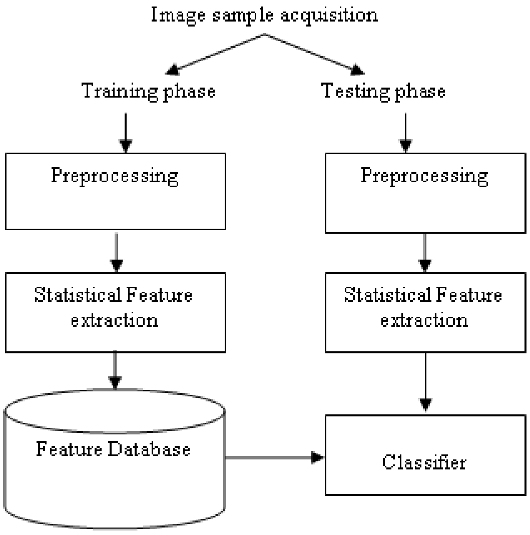


Figure 4. Block diagram of proposed methodology

2.1. Image Set

The set of 929 image samples of fruits affected by fungal disease symptoms are considered for the work. Fruits like mango (Mauls domestic), pomegranate (Punic granite) and grape (Vitas viniferous) are considered for the study. The chosen fungal disease symptoms affected on each fruit type are i)anthracnose, powdery mildew affected on mango, ii) anthracnose affected on pomegranate, iii) anthracnose, powdery mildew, downey mildew affected on grape. In this work, we have considered image samples of fungal disease affected on different parts of the plant like stem, leaf and fruit. These fungal affected image samples are further categorized into 246 partially affected, 168 moderately affected, 215 severely affected and 300 normal, which are considered for classification purpose. The scientific classification of fungal symptom affected on each fruit type along with affected part is shown in (Table.1).

2.2. Preprocessing

The single fungal affected fruit image is captured by analog camera. Then preprocessing steps applied over image. The preprocessing of image includes shade correction, removing artifacts, formatting, binarization and edge detection. Formatting deals with storage representation and setting the attributes of the image. This formatted image is used as input to the binarization and edge detection process. The preprocessing is done at two phases. In the first phase, input image is preprocessed for binarization and noise removal is done using median filtering. The filtered image is resized to a constant resolution of size 30x30. Further, the image is thinned and bounding box is generated. In the second phase, input image is preprocessed for edge detection using canny edge detector. An edge-detection filter imfilter is used to improve the appearance of blurred or anti-aliased images. The output of this phase is edge detected image. The preprocessed image of an input image sample for phase one and two is shown in (Figure.5). The preprocessing procedure is given in (Algorithm.1 & 2).

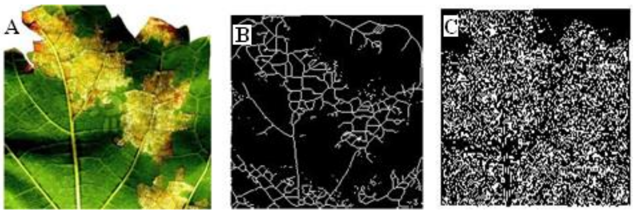


Figure 5. Preprocessed image: Input image (A); binary image (B); edge detected image(C)

```
Algorithm 1: Image acquisition and preprocessing
Input: Original 24-bit Color Image
Output: Binary image
Start
Step 1: Capture the image of fungal affected fruit using analog camera and save it in personal computer.
Step 2: Read input image from specified location.
Step 3: Apply shade correction on input image.
Step 4: Improve the quality of image by removing artifacts.
Step 5: Convert the input image into black and white image
Step 6: Filter the binary image using median filtering
Step 7: Resize the image to 30*30
Step 6: Generate thinned image
Step 7: Crop thinned image
Stop.
```

```
Algorithm 2: Image acquisition and preprocessing
Input: Original 24-bit Color Image
Output: Edge detected image
Start
Step 1: Capture the image of fungal affected fruit using analog camera and save it in personal computer.
Step 2:  Read input image from specified location.
Step 3: Apply shade correction on input image.
Step 4: Improve the quality of image by removing artifacts.
Step 5: Apply canny edge detector over input image.
Step 6: Filter the edge detected image using imfilter function.
Stop.
```

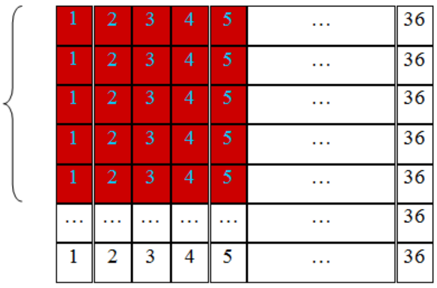
2.3. Feature Extraction

Features are the descriptors which specifies the different properties of an image for example color, size, shape, intensity, texture etc. Feature extraction is the processing of getting the statistical values from the image by some sort of calculations. In this work, we have used statistical based feature extraction methods for detection of fungal affected fruit.

Texture is an important characteristic of many natural surfaces and naturally occurring patterns. There are two widely used approaches to describe the texture of a region, namely statistical and structural. The statistical approaches considers that the intensities are generated by a two dimensional random field. The methods used are based on spatial frequencies and yield characterization of textures as smooth, coarse and grainy. Second order statistical texture features like GLCM and GLRM are used to carry out texture analysis. These methods are compared with first order block wise feature extraction method. First order statistics can be used as the most basic texture feature extraction methods, which are based on the probability of pixel intensity values occurring in digital images. The preprocessed image is given as input to block wise, GLCM, GLRM feature extraction methods.

2.3.1. Block wise feature extraction

The preprocessed image generated using Algorithm.1 is divided into various blocks each of size 5*5 as shown in (Figure.6). Then features are extracted from each row and column. The features are stored into a feature vector *F*. Totally 36 features are stored in feature vector *F*. The feature vector *F* is described in (Equation.1).



1	2	3	4	5	…	36
1	2	3	4	5	…	36
1	2	3	4	5	…	36
1	2	3	4	5	…	36
1	2	3	4	5	…	36
…	…	…	…	…	…	36
1	2	3	4	5	…	36

Figure 6. Image blocks

$$F = [f_i]; \quad 1 \le i \le 36 \qquad (1)$$

Where, f_i is feature vector of i^th block.

The block wise feature values of all blocks extracted from each row and column is shown in (Figure.7) for preprocessed image shown in (Figure.5 (B)).The block wise feature extraction procedure is given in (Algorithm.3).

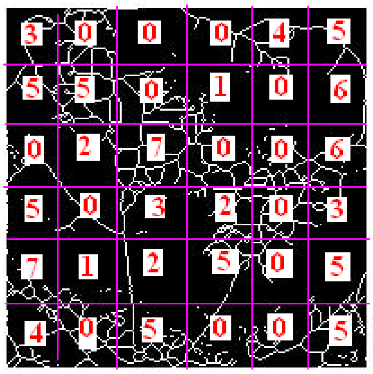


Figure 7. Row-column wise features for all blocks

F= [3 0 0 0 4 5 5 5 0 1 0 6 0 2 7 0 0 6 5 0 3 2 0 3 7 1 2 5 0 5 4 0 5 0 0 5]

```
Algorithm.3 : Block wise Feature Extraction
Input: Preprocessed Image (Output of Algorithm.1)
Output: Feature vector.
Start
Step 1: Divide the image into 5*5 blocks and find the number of pixels representing image in each block
Step 2: Find the size of image and store in variables, row and column
Step 3: Divide the value of row by 6
Step 4: Divide the value of column by 6
Step 5: Trace the row and column, start and end respectively for each block
Step 6: Extract the pixels in each block
Step 7: Count the number of each pixel and store into feature vector
Stop.
```

2.3.2. Gray Level Co-occurrence Matrix

The GLCM method of texture description is based on the repeated occurrence of some gray-level configuration in the texture. It was proposed by Haralick [33]. Haralick features calculation is done in two phases, i)Calculation of the Co-occurrence Matrices, ii)Calculation of the features based on the Co-occurrence Matrix. The textural features are evaluated using (Equations.2 to 10). The Co-occurrence Matrix is computed using (Algorithm.4). The textural feature extraction procedure is given in (Algorithm.5).

$$\text{Mean } (\mu) = \sum_{x} x \sum_{y} P(x,y) \qquad (2)$$

$$\text{Variance} = \sum_{x,y} (x-\mu)^2 P(x,y) \qquad (3)$$

$$\text{Range } = Max(p(x,y)) - min(p(x,y)) \qquad (4)$$

$$\text{Energy} = \sum_{i=1}^{Ng} \sum_{j=1} p^2 d(i,j) \qquad (5)$$

$$\text{Entropy} = -\sum_{I,J} P(i,j) \log P(i,j) \qquad (6)$$

$$\text{Homogeneity} = \sum_{i=1}^{Ng} \sum_{j=1}^{Ng} \frac{p_d(i,j)}{1+|i-j|} \qquad (7)$$

$$\text{Maximum Probability} = \max(P(x,y)) \qquad (8)$$

$$\text{Contrast} = \sum_{n=0}^{Ng-1} n^2 \sum_{|i-j|} Pd(i,j) \qquad (9)$$

$$\text{Inverse Difference Moment } = \sum_{x,y;x \neq y} \frac{P^{\lambda}(x,y)}{|x-y|^k} \qquad (10)$$

Where $\mu_x, \mu_y, \sigma_x, \sigma_y$ are means and standard deviations defined by (Equations.11 to 14).

$$\mu_x = \sum_{x} x \sum_{y} P(x,y) \qquad (11)$$

$$\mu_y = \sum_y y \sum_x P(x,y) \tag{12}$$

$$\sigma_x = \sum_x (x-\mu_x)^2 \sum_y P(x,y) \tag{13}$$

$$\sigma_y = \sum_y (y-\mu_x)^2 \sum_x P(x,y) \tag{14}$$

Algorithm. 4: Calculation of Co-occurrence Matrix $P_{f,\,d}(x, y)$ from the image f(x, y).

Input: Input gray level image f(x, y) (matrix of size M*N)

Output: Co-occurrence Matrix $P_{f,\,d}(x, y)$ for d=1 in the direction f.

Start

Step 1: Assign $P_{f,\,d}(x, y)$ =0 for all x, y Є [0, L], where L is the maximum gray level.

Step 2: For all pixels(x1, y1) in the image, determine (x2, y2), which is at distance d in direction f and perform $P_{f,\,d}$ [f(x1, y1), f(x2, y2)] = $P_{f,\,d}$ [f(x1, y1), f(x2, y2)] + 1

Stop.

Algorithm. 5: GLCM Textural Feature Extraction

Input: Preprocessed image (Output of Algorithm.2)

Output: Textural features

Start

Step 1: Derive the Gray Level Co-occurrence Matrices (GLCM) *Pφ, d (x, y)* for four different values of direction *φ* (*0^0, 45^0,90^0 and 135^0*) and d=*1* which are dependent on direction *φ*.

Step 2: Compute the Co-occurrence Matrix, which is independent of direction using *Algorithm.4*.

Step 3: GLCM features are calculated using Equations.(2) thru (10).

Stop.

We have found that only three features contribute as discriminating features as this is essential for better recognition and classification. Hence we have considered only variance, sum mean, and contrast as significant features. The reduced three GLCM texture features are shown in (Figure.8).

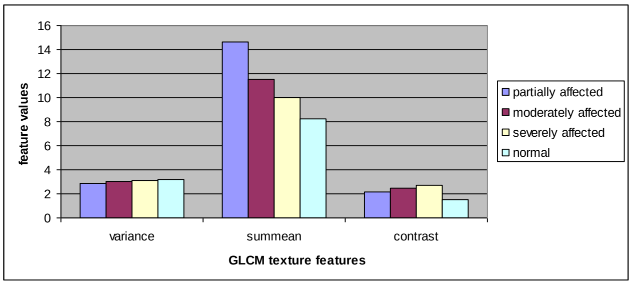


Figure 8. Texture features

2.3.3. Gray Level Run-length Matrix (GLRM)

Gray Level Run-length Matrix uses the basic idea of Run-length statistics for extracting such information from gray level runs of an image. Consecutive pixels of the same gray value or level, in a given direction, constitute a run. The number of runs of different lengths and gray values form a two dimensional matrix called Run-length matrix. An element of a RM, Q(x, y) represents the number of x gray values y is the considered Run-length [1].The feature extraction is done in two phases, i)Development of Run-length Matrix, ii)Calculation of features based on the Run-length Matrix. We have obtained from the Run-length matrix seven different texture features like Short Run Emphasis (SRE), Long Run Emphasis (LRE), Run Length Non-uniformity (RLN), Gray Level Non-uniformity (GLN), High Gray level Run-length Emphasis (HGRE), Low Gray level Run-length Emphasis (LGRE) and Run Percentage (RP). These features are obtained using (Equations.15 to 21). The Run-length Matrix is computed using (Algorithm.6). The textural feature extraction procedure is given in (Algorithm.7).

$$SRE = \frac{1}{n_r}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{P(i,j)}{j^2} \tag{15}$$

$$LRE = \frac{1}{n_r}\sum_{i=1}^{M}\sum_{j=1}^{N}P(i,j).j^2 \tag{16}$$

$$GLN = \frac{1}{n_r}\sum_{i=1}^{M}\left(\sum_{j=1}^{N}P(i,j)\right)^2 \tag{17}$$

$$RLN = \frac{1}{n_r}\sum_{i=1}^{N}\left(\sum_{j=1}^{M}P(i,j)\right)^2 \tag{18}$$

$$RP = \frac{n_r}{n_p} \tag{19}$$

$$LGRE = \frac{1}{n_r}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{P(i,j)}{i^2} \tag{20}$$

$$HGRE = \frac{1}{n_r}\sum_{i=1}^{M}\sum_{j=1}^{N}P(i,j).i^2 \tag{21}$$

Algorithm 6: Development of Run-length Matrix Qφ (x, y) from the Image f(x, y).

Input: Gray level image f(x, y) of size M*N

Output: Run-length matrix Qφ(x, y) in the direction φ.

Start

Step 1: Assign Qφ(x, y) =0 for all x, y ε [0, L], L is the maximum gray level.

Step 2: Find the matrix Qφ(x, y), for a given angle φ. The entry Q(x, y) is the (x, y)th entry in the Run-length matrix, where 'x' is the gray level and 'y' is the Run-length.

Stop.

Algorithm 7: GLRM Texture Feature Extraction

Input: Preprocessed image (Output of Algorithm.2)

Output: Textural features

Start

Step 1: Derive the Run-length Matrices Qφ(x, y) for four different directions φ (00, 450, 900 and 1350).

Step 2: Compute the Run-length matrix, independent of direction using the equations (15) thru (21)

Step 3: Run-length matrix features are calculated using Equations.(15) thru (21).

Stop.

We have found that only two features contribute as discriminating features as this is essential for better recognition and classification. Hence we have considered only Run Length Non-uniformity (RLN), Gray Level Non-uniformity (GLN) as significant features. The reduced two GLRM texture features are shown in (Figure.9).

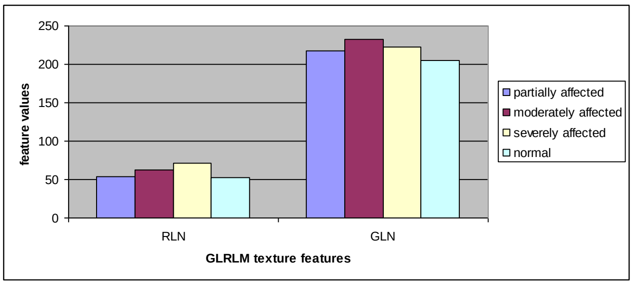


Figure 9. Texture features

2.4. Classifier

We have used a Nearest Neighbor classifier for classification purpose. This is amongst the simplest of all classification algorithms in supervised learning. This is a method of classifying patterns based on the class label of the closest training patterns in the feature space. There is no training time required for this classifier. Every time a test pattern is to be classified, it has to be compared with all the training patterns, to find the closest pattern. The classification is done according to some similarity of the test pattern to the training patterns. To determine this similarity/dissimilarity, proximity measures are used. The distance between two patterns is used as a proximity measure. The Euclidean distance is the most popular distance measure. This is because Euclidean distance is easy for human comprehension, rotation and translation invariant.

The training phase constitutes calculation of statistical features extracted from fungal affected fruit image samples. The extracted features are stored into database. The classifier is tested on the test images for each class. The classifier is based on the Euclidean distance from the feature vector representing the test image and every record in feature database using (Equation.22). The classifier used the Nearest Neighbor principle.

$$\text{Distance}(Test, Train) = \sqrt{\sum\left(DTest - TrainD\right)^2} \qquad (22)$$

The test image is classified as belonging to a particular class to which its Euclidean distance is minimum among the calculated distances.

3. Results and Discussion

All the algorithms used in this work are implemented using MATLAB 7.0. The image samples are divided into two halves and one half is used for training and other is for testing. The percentage accuracy is defined as the ratio of correctly recognized image samples to the total number of test image samples. The Percentage accuracy is given by (Equation.23).

Percentage accuracy (%) =

$$\frac{\text{Correctly Recognized Image Samples}}{\text{Total Number of Test Image Samples}} * 100. \qquad (23)$$

The individual average classification accuracy based on disease severity levels is shown in (Figure.10). The highest recognition and classification accuracy of 98.76% is observed with severely affected and the lowest of 88.32% is observed with normal using block wise feature extraction. The highest recognition and classification accuracy of 95% is observed with severely affected and the lowest of 85% is observed with normal using GLCM feature extraction. The highest recognition and classification accuracy of 92% is observed with severely affected and the lowest of 81.33% is observed with normal using GLRM feature extraction.

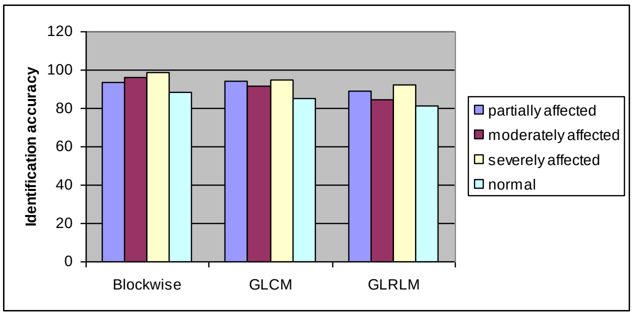


Figure 10. Classification accuracy based on disease severity levels

The average classification accuracies using feature extraction methods block wise, GLCM, GLRM are 94.085%, 91.37% and 86.715% for fungal affected fruits' image samples is shown in (Figure.11).

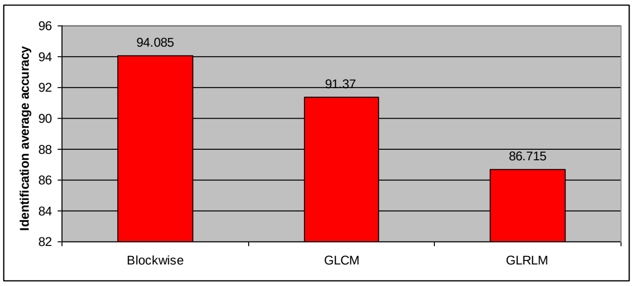


Figure 11. Average classification accuracy for each feature type

The image samples are selected randomly for training and testing. For each training and testing, experimentation is performed 10 times (trials) and classification is calculated for each time. The minimum classification, maximum classification and average classification accuracy obtained across 10 trials is shown in (Figure.12).

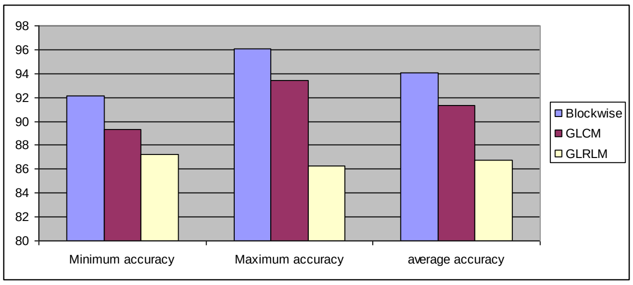


Figure 12. Classification results using statistical features

The GLCM and GLRM computes properties of the image related to second-order statistics which considers the relationship among pixels or groups of pixels, whereas, fungal affected areas have no spatial relationship among pixels and hence block wise features related to first order statistics will give better results than GLCM and GLRM.

4. Conclusion

We have developed statistical methodologies for detection of fruits' image samples affected by fungal disease based on disease severity levels. The evaluation of statistical features like block wise, GLCM and GLRM is done. The classification is performed using nearest neighbor with Euclidean distance. The work finds application in automatic recognition and classification of disease affected on fruits by the service robots in the real world.

For future study, further different neural network architectures, SVM, fuzzy based classifiers can be used for classification. We can extend this work to classify fungal disease symptoms affected on commercial crops, cereals, vegetables .The work can also be extended to identify various diseases like viral, bacterial affected on agriculture/horticulture produce.

Acknowledgements

The authors wish to thanks University of Horticulture sciences, Bagalkot, INDIA and department of horticulture, University of Agricultural sciences, Dharwar, INDIA for providing test images used in this work. We would like to express our thanks to Dr.B.S.Anami, principal, KLE.Institute of Technology, Hubli, INDIA for his valuable suggestions.

References

[1] Jagadeesh D. Pujari, Rajesh.Yakkundimath and A.S.Byadgi (2013). Grading and Classification of Anthracnose Fungal Disease in Fruits. International Journal of Advanced Science and Technology. Vol.52.

[2] M.Akhil.Jabbar, D.L.Deekshatulu and Priti.Chandra (2013). Heart Disease Classification using Nearest Neighbor Classifier with Feature Subset Selection. Annals Computer Science Series. 11th Tome 1st Fasc.

[3] Vishal S.Thakare, Nitin N. Patil and Jayshri S. Sonawane (2013). Survey on Image Texture Classification Techniques. International Journal of Advancements in Technology. Vol.4. Page.1.

[4] Sudheer reddy bandi, A Varadharajan and A Chinnasamy (2013). Performance evaluation of various Statistical Classifiers in Detecting the Diseased Citrus Leaves. International Journal of Engineering Science and Technology (IJEST). Vol. 5.

[5] Shiv Ram Dubey and Anand Singh Jalal (2012). Detection and Classification of Apple Fruit Diseases using Complete Local Binary Patterns. Third IEEE International Conference on Computer and Communication Technology (ICCCT), Allahabad, India. Pages. 346-351.

[6] Shiv Ram Dubey and Anand Singh Jalal (2012). Adapted Apple for Fruit Disease Identification using Images. International Journal of Computer Vision and Image Processing (IJCVIP). Vol. 2. Pages. 51 – 65.

[7] J.K.Patil and Raj Kumar (2012). Feature Extraction of Diseased Leaf Images. Journal of signal and image processing. Vol.3. Pages.60-63.

[8] R. Mishra, D. Karimi, R. Ehsani and W. S. Lee (2012). Identification of Citrus Greening using a VIS-NIR Spectroscopy Technique. Transactions of the ASABE. Vol. 55. Pages. 711-720.

[9] Mandeep Singh and Bharti Chauhan (2012). Classification: A holistic View. International Journal of Computer Science & Communication. Vol. 3. Pages. 69-72.

[10] Sabine D. Bauer, Filip Korc and Wolfgang Forstner (2011). The Potential of Automatic Methods of Classification to Identify Leaf Diseases from Multispectral Images. Precision Agriculture, Springer-Verlag. Vol.12.Pages.361-377.

[11] Jayamala K. Patil and Raj Kumar (2011). Advances in Image processing for Detection of Plant Diseases. Journal of Advanced Bioinformatics Applications and Research. Vol.2. No.2. Pages. 135-141.

[12] N.A.Lili, F. Khalid and N.M. Borhan(2011).Classification of Herbs Plant Diseases via Hierarchical Dynamic Artificial Neural Network after Image Removal using Kernel Regression Framework. International Journal on Computer Science and Engineering. Vol. 3. No.1.

[13] D. Moshou, C. Bravo, R. Oberti, J.S. West, H. Ramon, S. Vougioukas and D. Bochtis (2011). Intelligent Multi-sensor system for the Detection and Treatment of Fungal Diseases in Arable Crops. Biosystems Engineering. Vol.18. Pages. 3 11 -3 2 1.

[14] D.S.Guru, P.B.Mallikarjuna and S.Manjunath (2011). Segmentation and Classification of Tobacco Seedling Diseases. COMPUTE '11 Proceedings of the Fourth Annual ACM Conference, Bangalore, India.

[15] Basvaraj S. Anami, J.D.Pujari and Rajesh.Yakkundimath (2011). Identification and Classification of Normal and Affected Agriculture/horticulture Produce based on Combined Color and Texture Feature Extraction. International Journal of Computer Applications in Engineering Sciences. Vol.1.

[16] H. Al-Hiary, S. Bani-Ahmad, M. Reyalat, M. Braik and ALRahamneh (2011). Fast and Accurate Detection and Classification of Plant Diseases. International Journal of Computer Applications. Vol.17.

[17] Ryusuke Nosaka, Yasuhiro Ohkawa, and Kazuhiro Fukui (2011). Feature Extraction Based on Co-occurrence of Adjacent Local Binary Patterns. Springer-Verlag, Berlin Heidelberg, Part II, LNCS 7088. Pages. 82–91.

[18] Di Cui, Qin Zhang, Minzan Li, Glen L.Hartman and Youfu Zhao (2010). Image processing Methods for Quantatively Detecting Soybean Rust from Multispectral Images. Biosystems Engineering. Vol.107. Pages.186-193.

[19] T.Rumpf, A.K.Mahlein, U.Steiner, E.C.Oerke, H.W.Dehne and L.Plumer (2010). Early Detection and Classification of Plant Diseases with Support Vector Machines based on Hyperspectral Reflectance. Computers and Electronics in Agriculture. Vol.74. Pages.91-99.

[20] A. Camargo and J.S. Smith (2009). Image Pattern Classification for the Identification of Disease causing Agents in Plants. Computers and Electronics in Agriculture. Vol.66, Page.121–125.

[21] Qing Yao, Zexin Guan, Yingfeng Zhou, Jian Tang, Yang Hu and Baojun Yang (2009). Application of Support Vector Machine for Detecting Rice Diseases using Shape and Color Texture Features. International Conference on Engineering Computation.

[22] Di Cui, Qin Zhang, Minzan Li, Glen L.Hartman and Youfu Zhao (2009). Detection of Soybean Rust using a Multispectral Image Sensor. Sensor & Instruments, Food Quality, Springer-Verlag. Vol.3. Pages.49-56.

[23] Dae Gwan Kim, Thomas F. Burks, Jianwei Qin and Duke M. Bulanon. (2009). Classification of Grapefruit Peel Diseases using Color Texture Feature Analysis. International journal of Agricultural & Biological Engineering. Vol. 2.

[24] Geng Ying, Li Miao, Yuan Yuan and Hu Zelin (2008). A Study on the Method of Image Preprocessing for Recognition of Crop Diseases. International Conference on Advanced Computer Control.

[25] Abdul Malik Khan and Andrew P. Papliński (2008). Blemish Detection in Citrus Fruits. Proceedings of SPIT-IEEE Colloquium and International Conference, Mumbai, India. Vol.1.Pages.203-211.

[26] Kuo-Yi Huang (2007). Application of Artificial Neural Network for Detecting Phalaenopsis Seedling Diseases using

Color and Texture features. Computers and Electronics in agriculture. Vol.57. Pages.3–11.

[27] Alexander A. Doudkin, Alexander V. Inyutin, Albert I. Petrovsky and Maxim E. Vatkin (2007). Three-level Neural Network for Data Clusterization on Images of Infected Crop Field. Journal of Research and Applications in Agricultural Engineering. Vol. 52.

[28] R. Pydipati, T.F. Burks and W.S. Lee (2006). Identification of Citrus Disease using Color Texture Features and Discriminate Analysis. Computers and Electronics in Agriculture. Vol.52. Pages. 49-59.

[29] Hamid Muhammed and Hamed (2005). Hyperspectral Crop Reflectance Data for Characterizing and Estimating Fungal Disease Severity in Wheat, Biosystems Engineering. Vol.91. Pages.9-20.

[30] D.G. Sena Jr, F.A.C. Pinto, D.M. Queiroz and P.A. Viana (2003). Fall Armyworm Damaged Maize Plant Identification using Digital Images. Biosystems Engineering. Vol.85. Pages.449–454

[31] Pinstrup-Andersen (2001). The Future World Food Situation and the Role of Plant Diseases. DOI: 10.1094/PHI-I-2001-0425-01.

[32] Marc Lefebvre, Sylvia Gil, Denis Brunet, E. Natonek, C. Baur, P. Gugerli and Thierry Pun(1993). Computer Vision and Agricultural Robotics for Disease Control: the Potato operation. Computers and Electronics in Agriculture. Vol.9. Pages. 85-10.

[33] R.M.Harlick, K.shanmugam and H.Dinstein (1973). Texture Features for Image Classification. IEEE Transactions on systems, man and cybernetics. Vol.3. Pages.610-621.

[34] Rafael C. Gonzalez (2009). Digital Image Processing using MATLAB, second edition, Addison-Wesley Publishing Company.

[35] R. O. Duda, P. E. Hart and D. G. Stork (2000). Pattern Classification, John Wiley & Sons.

[36] Dr. K T. Chandy. Important Fungal Diseases: Plant Disease Control, Booklet No. 342,PDCS.4.

[37] C. H. Chen, L. F. Pau and P. S. P. Wang (1998).The Handbook of Pattern Recognition and Computer Vision, second edition, Pages. 207-248, World Scientific Publishing Corporation.