Some Experiments on Fitting of Gielis Curves by Simulated Annealing and Particle Swarm Methods of Global Optimization

SK Mishra
Dept. of Economics
NEHU, Shillong (India)

Introduction: The Gielis superformula

$$r(\theta) = f(\theta).\left[\left|\tfrac{1}{a}\cos(\tfrac{m}{4}\theta)\right|^{n_2} + \left|\tfrac{1}{b}\sin(\tfrac{m}{4}\theta)\right|^{n_3}\right]^{(-n_1^{-1})} = f(\theta).g(\theta)\ ;\ m>0 \qquad \ldots (1)$$

describes almost any closed curve in terms of the deformed circle (or ellipse), $g(\theta)$, and another function, $f(\theta)$, and their parameters (Gielis, 2003). The function $f(\theta)$ may be considered as a modifier of the Gielis function, $g(\theta)$.

Estimation of Gielis Parameters: For a scientific purpose, Gielis parameters need to be estimated from empirical data. Presently, we are concerned with the possibilities of the same. Let the n true points be $[z_i = (x_i, y_i);\ i = 1,2,...,n]$, of which the corresponding observed values are $z' = (x_i', y_i')$, possibly with errors of measurement and displacement of origin by (c_x, c_y), unknown to us. Let $(\tilde{c}_x, \tilde{c}_y)$ be the approximate or assumed values of (c_x, c_y). Let us denote by $\tilde{z}_i = (\tilde{x}_i, \tilde{y}_i) = (x_i' - \tilde{c}_x, y_i' - \tilde{c}_y)$. From these values we obtain $\tilde{r}_i = \sqrt{(\tilde{x}_i^{\,2} + \tilde{y}_i^{\,2})}$. We also obtain $\tilde{\theta}_i = \tan^{-1}(\tilde{y}_i / \tilde{x}_i)$. On the other hand, we obtain $\hat{r}_i = g(\tilde{\theta}_i, \tilde{a}, \tilde{b}, \tilde{m}, \tilde{n}_1, \tilde{n}_2, \tilde{n}_3).f(\tilde{\theta})$, where $g(.)$ is the Gielis super-formula defined in (4) and $f(\theta)$ is variously defined. The wavy bar on the arguments of $g(.)$ and $f(.)$ indicates that all parameters have taken on some arbitrary values, which may not be the correct values. The deviation of assumed values of parameters from their true values gives rise to $d_i = abs(\tilde{r}_i - \hat{r}_i)$ and consequently $S^2 = \sum_{i=1}^{n} d_i^2 \geq 0$. Only if the assumed values of parameters are the true values, S^2 can be zero, but smaller it is, closer are the assumed values of the parameters from their true values (assuming empirical uniqueness of the parameters to a given set of data). Thus we have to find the values of Gielis parameters in $g(.)$ and $f(.)$ such that S^2 is minimum.

Minimization of S^2 poses formidable problems due to two reasons. First, the Gielis parameters are not unique to data, suggesting that minima (local as well as global) are located in the valleys or deep trenches. The parameters of the deformed circle, $g(\theta)$, and the modifying function, $f(\theta)$, interact among themselves. A large number of experiments carried out by the author make the basis of this view. Secondly, the parameters span a highly nonlinear surface of S^2, which has innumerably many local minima (Mishra, 2006).

- Working Paper (July, 2006): Dept. of Economics, NEHU, Shillong (India) – 793022
- Acknowledgement: The author is indebted to Professor P.N. Suganthan (*School of Electrical and Electronic Engineering, Nanyang Technological University, 639798 Singapore)* for generous help by sending his papers to the author.

As it is well known, most of the nonlinear optimization procedures that were developed in the 1960's or before are extremely prone to be caught in the local optima if the surface to be optimized is substantially irregular, ridged and multi-modal. In the due course, researchers in the field of operations research turned to learning from nature and imitating the schemes in which natural processes attain a minimum. Understanding the process of adaptation of living beings to their environment for a survival led to development of the 'Genetic Algorithm' (Holland, 1975) and the optimization method based on adaptation (Goldberg, 1989; Wright, 1991). This method mimics the process of survival of the fittest. Another very important and effective method - the Particle Swarm method (Eberhart and Kennedy, 1995; see Parsopoulos and Vrahatis, 2002) - was motivated by the behaviour of birds, fish and insects. On the other side, researchers learned from physics – the process of annealing in metallurgy (Kirkpatrick et al., 1983) and the method of 'simulated annealing' was developed

The Simulated Annealing Method of Global Optimization: The simulated annealing method mimics the annealing process in metallurgy. In an annealing process a metal in the molten state (at a very high temperature) is slowly cooled so that the system at any time is approximately in thermodynamic equilibrium. As cooling proceeds, the system becomes more ordered – the liquid freezes or the metal re-crystallizes – attaining the ground state at T=0. This process is simulated through the Monte Carlo experiment (Metropolis et al. 1953). If the initial temperature of the melt is too low or cooling is done unduly fast the metal may become 'quenched' due to being trapped in a local minimum energy state (meta-stable state) forming defects or freezing out.

The simulated annealing method of optimization makes very few assumptions regarding the function to be optimized, and therefore, it is quite robust with respect to non-quadratic surfaces. In this method, the mathematical system describing the problem mimics the thermodynamic system. The current solution to the problem mimics the current state of the thermodynamic system, the objective function mimics the energy equation for the thermodynamic system, and the global minimum mimics the ground state (Kirkpatrick et al., 1983; Cerny, 1985). However, nothing in the numerical optimization problem directly mimics the temperature, T, in the thermodynamic system underlying the metallurgical process of annealing. Therefore, a complex abstraction mimics it. An arbitrary choice of initial value of a variable called 'temperature', how many iterations are performed at each 'temperature', the step length at which the decision variables are adjusted, and the rate of fall of 'temperature' at each step as 'cooling' proceeds together make an 'annealing schedule'. This schedule mimics the cooling process. At a high 'temperature' the step lengths at which the decision variables are adjusted are larger than those at a lower 'temperature'. Whether the system is trapped into local minima (quenching takes place) or it attains the global minimum (faultless crystallization) is dependent on the said annealing schedule. A wrong choice of the initial 'temperature', or the rate of fall in the 'temperature' leads to quenching or entrapment of the solution in the local minima. The method does not provide any clear guideline as to the choice of the 'annealing schedule' and often requires judgment or trial and error. If the schedule is properly chosen, the process attains the global minimum.

Particle Swarm Method of Global Optimization: A swarm of birds or insects or a school of fish searches for food, protection, etc. in a very typical manner. If one of the members of the swarm sees a desirable path to go, the rest of the swarm will follow up quickly. Every member of the swarm searches for the best in its locality - learns from its own experience. Additionally, each member learns from the others, typically from the best performer among them. Even human beings show a tendency to learn from their own experience, their immediate neighbours and the ideal performers.

The Particle Swarm method of optimization mimics the said behaviour (see Wikipedia : http://en.**wikipedia**.org/wiki/Particle_**swarm**_optimization). Every individual of the swarm is considered as a particle in a multidimensional space that has a position and a velocity. These particles

fly through hyperspace and remember the best position that they have seen. Members of a swarm communicate good positions to each other and adjust their own position and velocity based on these good positions. There are two main ways this communication is done: (i) "swarm best" that is known to all (ii) "local bests" are known in neighborhoods of particles. Updating the position and velocity is done at each iteration as follows:

- $x \leftarrow x + v$
- $v \leftarrow wv + c_1 r_1(\hat{x} - x) + c_2 r_2(\hat{x}_g - x)$
  - w is the inertial constant. Good values are usually slightly less than 1.
  - c_1 and c_2 are constants that say how much the particle is directed towards good positions. Good values are usually right around 1.
  - r_1 and r_2 are random values in the range [0,1].
  - $\hat{x}$ is the best the particle has seen.
  - $\hat{x}_g$ is the global best seen by the swarm. This can be replaced by $\hat{x}_l$, the local best, if neighborhoods are being used.

The Particle Swarm method has many variants. Among them, the Repulsive Particle Swarm (RPS) method of optimization (see Wikipedia, http://en.wikipedia.org/wiki/RPSO) is particularly effective in finding out the global optimum in very complex search spaces (although it may be slower on certain types of optimization problems). Other variants use a dynamic scheme (Liang and Suganthan, 2005; Huang et al., 2006).

In RPS the future velocity, v_{next} of a particle at position $\mathbf{x}$ with a recent velocity $\mathbf{v}$ is calculated by

$$\mathbf{v}_{\text{next}} = \omega\mathbf{v} + a\ \chi_1\ (-\mathbf{x} + \hat{\mathbf{x}}) + b\ \chi_2\ \omega(-\mathbf{x} + \hat{\mathbf{y}}) + c\ \chi_3\ \omega\ \mathbf{z}$$

where,

- χ_1, χ_3, χ_3 : random numbers $\in (0,1)$
- ω : inertia weight $\in (0.01, 0.7)$
- $\hat{\mathbf{x}}$: best position of a particle
- $\hat{\mathbf{y}}$: best position of a randomly chosen other particle from within the swarm
- $\mathbf{z}$: a random velocity vector
- a,b,c : constants

The future x that is, x_{next} is defined as $x_{next} = x + v_{next.}$ Occasionally, when the process is caught in a local optimum, some perturbation of v may be needed.

The Simulation Experiments: We have experimented with nine different models. All these models are instances of a deformed circle, $g(.)$, modified by different modifier functions, $f(.)$. Three typical instances of $g(.)$ have been chosen. The parameters of $g(.)$ are given in table A.1. Three typical modifier functions are chosen, as given below. The chosen values of n_4 and n_5 are also given in table-A.1.

$f_1(\theta) = r = [n_4(3\cos(t) - \cos(3t))^2 + n_5(3\sin(t) - \sin(3t))^2]^{0.5}$: (Nephroid) … (2)

$f_2(\theta) = \mathrm{r} = \mathrm{n}_4 + \mathrm{n}_5 \cos(\mathrm{t})$: (Limaçon) … (3)

$f_3(\theta) = \mathrm{r} = \mathrm{n}_4 - \mathrm{n}_5 \cos(\mathrm{t}) + \mathrm{abs}(\cos(\mathrm{t}))^3$ …(4)

In all the three modifier functions, n_4 and n_5 are parameters and $0 \le t \le 2\pi$.

In case of each model, hundred uniformly distributed random points have been generated with the parameters specified in the relevant $g(.)$ and $f(.)$. The Classical simulated annealing (CSA of Kirkpatrick et al., 1983) and the Repulsive Particle Swarm methods of optimization (RPS) have been repetitively applied to estimate the parameters. The CSA method requires the bounds (the lower and the upper limits; LL and UL) on the parameters to be specified. For all the nine models we have used the identical set of bounds, specified in table-A.1. However, RPS does not require such bounds. The jointly estimated parameters of $g(.)$ and $f(.)$ by both methods are presented in table-A.1. Their graphs are presented in Fig.A.1 $M_{ij}(i, j = 1, 2, ..., 3)$. The red points are those generated by the true parameters, the blue ones are generated by using the RPS-estimated parameters. The CSA-generated points are not plotted just to avoid clumsiness. For each model, the RPS-estimated (blue) points are superimposed on the generated (red) points to facilitate a visual assessment of the quality of fit, which is quantitatively represented by the value of S^2.

The Findings: The CSA as well as RPS method of global optimization performs very well in fitting the Gielis curves to data. The CSA often performs better than the RPS. In general, increase in the number of iterations to obtain the estimates improves performance of these methods greatly. So we do not intend to conclude as to the relative performance of these two methods in general. Our observations are limited to the present context only.

The Repulsive Particle Swarm program (written by the author in FORTRAN) converges much faster (than the CSA program). It does not require limits on the decision variables either. The initial guesses of the decision variables may simply be generated randomly, lying between -0.5 to +0.5 or so. These advantages may add to the applicability of RPS in finding global optima of complex multi-modal nonlinear optimization problems.

Extrapolation: How reliably can we extrapolate the points on the Gielis's $g(.)$ beyond the sample points used in estimating the parameters of $\hat{g}(.)$? Extrapolation has many applications such as shape recovery, etc. (Bhabhrawala and Krovi, 2005). For this exercise, we generated 100 points on $g(.)$ randomly and used only 70 of them (red points in Fig.A.2) to fit the Gielis's $\hat{g}(.)$ and the rest 30 points (green ones in Fig. A.2) were left out. With the estimated parameters (Table A.2) we generated 1000 points (blue ones in Fig.A.2) randomly strewn over the entire curve. The scatter shows that the estimated parameters can very well predict the points beyond the samples used for estimation. The values of S^2 in all the three cases are quite small.

References

- Bhabhrawala, T. and Krovi, V.: "Shape Recovery from Medical Image Data using Extended Superquadrics", *ASME 2005 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Long Beach, California USA, September 24-28, 2005.
- Cerny, V., "Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm", *J. Opt. Theory Appl.*, 45, 1, 41-51, 1985
- Eberhart R.C. and Kennedy J.: "A New Optimizer using Particle Swarm Theory", *Proceedings Sixth Symposium on Micro Machine and Human Science*, pp. 39–43. IEEE Service Center, Piscataway, NJ, 1995.
- Gielis, J.: "A Generic Geometric Transformation that unifies a Wide Range of Natural and Abstract Shapes", *American Journal of Botany,* 90(3): pp. 333–338, 2003.
- Goldberg, D.E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley, Reading, Mass, 1989.
- Holland, J.: *Adaptation in Natural and Artificial Systems,* Univ. of Michigan Press, Ann Arbor, 1975.
- Huang, V.L., Suganthan, P.N. and Liang, J.J. "Comprehensive Learning Particle Swarm Optimizer for Solving Multiobjective Optimization Problems", *International Journal of Intelligent Systems*, 21, pp.209–226 (Wiley Periodicals, Inc. Published online in Wiley InterScience www.interscience.wiley.com) , 2006
- Kirkpatrick, S., Gelatt, C.D. Jr., and Vecchi, M.P.: "Optimization by Simulated Annealing", *Science*, 220, 4598, 671-680, 1983.
- Liang, J.J. and Suganthan, P.N. "Dynamic Multi-Swarm Particle Swarm Optimizer", *International Swarm Intelligence Symposium*, IEEE # 0-7803-8916-6/05/$20.00. pp. 124-129, 2005. (obtained through personal request made by the author to epnsugan@ntu.edu.sg).
- Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E.: "Equation of State Calculations by Fast Computing Machines", *J. Chem. Phys.*,21, 6, 1087-1092, 1953.
- Mishra, S.K.: "On Estimation of the Parameters of Gielis Superformula from Empirical Data" *Social Science Research Network* (SSRN): http://ssrn.com/abstract=905051, Working Paper Series, 2006.
- Mishra, S.K.: "Experiments on Estimation of the Parameters of Gielis Super-Formula by Simulated Annealing Method of Optimization" *Social Science Research Network* (SSRN): http://ssrn.com/abstract=910800 , 2006.
- Parsopoulos, K.E. and Vrahatis, M.N., "Recent Approaches to Global Optimization Problems Through Particle Swarm Optimization", *Natural Computing*, 1 (2-3), pp. 235-306, 2002.
- Wright, A.H.: "Genetic Algorithms for Real Parameter Optimization", in GJE Rawlings (ed) *Foundations of Genetic Algorithms*, Morgan Kauffmann Publishers, San Mateo, CA, pp. 205-218, 1991.

Appendix

Table-A.1. True & Estimated Gielis Parameters (Modified Curves) with Limits on them

	c_x	c_y	a	b	n_1	n_2	n_3	m	n_4	n_5	S^2	
M # 11	0	0	1	1	0.6	2	3	3	3	2	0	T
	-15	-15	0	0	-80	-80	-80	0	-10	-10	LL	
	15	15	50	50	80	80	80	80	10	10	UL	
	0.010	-0.082	0.763	27.987	3.859	2.677	-0.089	6.022	5.898	3.918	0.435	C
	0.025	-0.087	-0.847	6.166	3.417	2.296	-0.163	6.006	5.996	4.048	0.508	P
M # 12	0	0	1	1	2	2	6	5	3	2	0	T
	-15	-15	0	0	-80	-80	-80	0	-10	-10	LL	
	15	15	50	50	80	80	80	80	10	10	UL	
	-0.005	0.088	0.568	0.831	9.809	3.550	-0.290	10.024	4.714	3.155	0.952	C
	-0.007	0.094	5.856	-0.969	-8.170	0.913	3.762	10.034	4.570	3.065	0.960	P
M # 13	0	0	1	1	8	4	-4	6	3	2	0	T
	-15	-15	0	0	-80	-80	-80	0	-10	-10	LL	
	15	15	50	50	80	80	80	80	10	10	UL	
	0.000	0.000	0.208	2.830	20.109	9.940	-10.062	6.000	8.484	5.652	0.001	C
	0.0014	0.001	3.885	-1.631	6.134	2.884	-3.083	6.004	4.894	3.280	0.026	P
M # 21	0	0	1	1	0.6	2	3	3	3	2	0	T
	-15	-15	0	0	-80	-80	-80	0	-10	-10	LL	
	15	15	50	50	80	80	80	80	10	10	UL	
	0.040	0.015	0.951	0.472	-63.643	-4.760	19.382	6.077	3.091	2.029	0.264	C
	3.3468	0.096	-0.388	1.845	7.971	2.597	-0.123	-4.129	4.354	-2.072	0.094	P
M # 22	0	0	1	1	2	2	6	5	3	2	0	T
	-15	-15	0	0	-80	-80	-80	0	-10	-10	LL	
	15	15	50	50	80	80	80	80	10	10	UL	
	-0.063	0.049	0.410	2.496	13.073	3.808	-0.537	9.940	3.914	2.707	0.315	C
	1.639	0.061	-0.776	-0.782	10.096	12.156	6.088	-3.963	4.208	0.509	0.790	P
M # 23	0	0	1	1	8	4	-4	6	3	2	0	T
	-15	-15	0	0	-80	-80	-80	0	-10	-10	LL	
	15	15	50	50	80	80	80	80	10	10	UL	
	0.000	0.000	43.724	10.289	-58.308	42.069	29.224	6.000	9.624	6.415	0.005	C
	0.004	0.010	6.149	-1.034	2.049	1.722	-1.013	-6.009	3.024	2.024	0.019	P
M # 31	0	0	1	1	0.6	2	3	3	3	2	0	T
	-15	-15	0	0	-80	-80	-80	0	-10	-10	LL	
	15	15	50	50	80	80	80	80	10	10	UL	
	-0.982	-0.039	0.949	5.796	15.318	4.294	1.637	5.959	3.381	1.294	0.504	C
	-0.541	-0.024	0.703	-1.693	5.921	3.156	0.105	-5.934	4.098	2.000	0.619	P
M # 32	0	0	1	1	2	2	6	5	3	2	0	T
	-15	-15	0	0	-80	-80	-80	0	-10	-10	LL	
	15	15	50	50	80	80	80	80	10	10	UL	
	1.217	0.027	13.680	50.000	50.370	11.819	10.236	8.052	7.876	2.254	2.371	C
	-0.092	0.0038	-0.747	-0.046	6.826	2.767	-0.154	-9.953	3.636	2.267	0.728	P
M # 33	0	0	1	1	8	4	-4	6	3	2	0	T
	-15	-15	0	0	-80	-80	-80	0	-10	-10	LL	
	15	15	50	50	80	80	80	80	10	10	UL	
	0.000	0.000	0.004	1.003	-78.492	-21.839	39.256	6.000	2.995	1.996	0.006	C
	0.005	0.047	0.748	2.075	3.956	11.822	1.479	5.985	4.018	2.695	1.132	P

M=Model; LL=Lower Limits; UL=Upper Limits; T=True Parameters; C and P are Estimated Parameters : C = Classical Simulated Annealing method; P = Repulsive Particle Swarm Method

Figures-A.1. (Plots of Generated and RPS-estimated points)

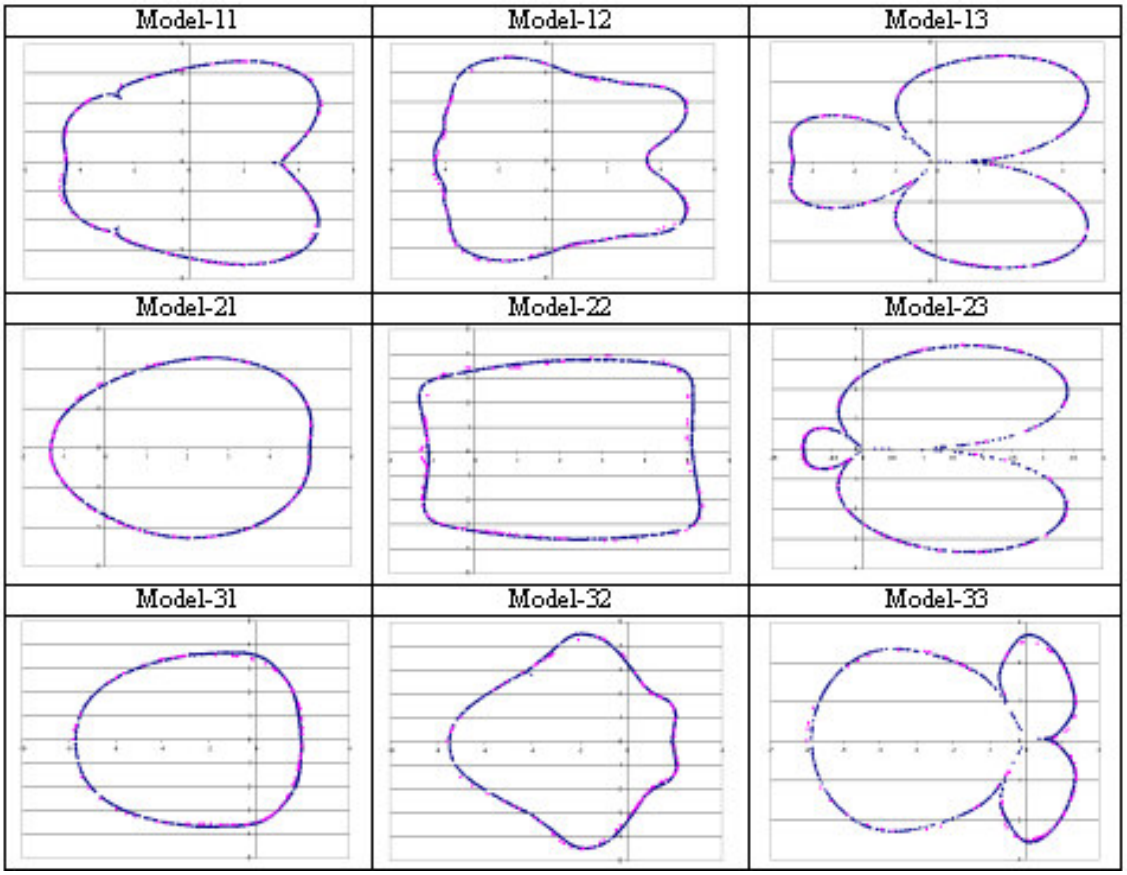
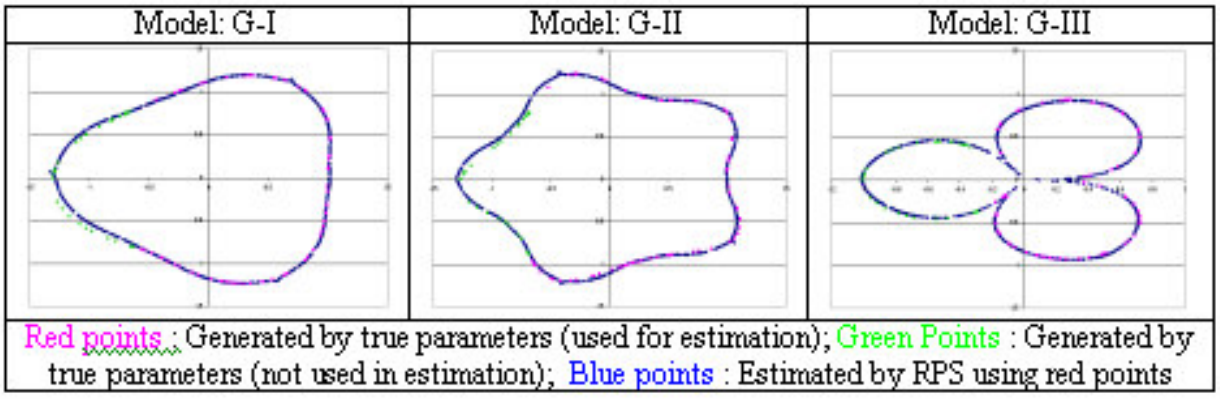


Table-A.2. True (T) and Estimated (E) Parameters of Gielis's g(.) function by RPS Method

Parameters			c_x	c_y	a	b	n_1	n_2	n_3	m	s^2
MCLEL	G I	T	0.0	0.0	1.0	1.0	0.6	2.0	3.0	3.0	0.0
		E	0.0145	-0.0023	1.1282	0.6544	-6.1967	-0.1674	2.9399	-6.0949	0.0035
	G II	T	0.0	0.0	1.0	1.0	2.0	2.0	6.0	5.0	0.0
		E	-0.0155	0.0138	1.6507	0.6637	-7.5254	-0.1936	3.5724	-9.9795	0.0268
	G III	T	0.0	0.0	1.0	1.0	8.0	4.0	-4.0	6.0	0.0
		E	0.0050	-0.0001	-1.8626	0.9682	3.9035	1.3954	-1.9119	6.0284	0.0170

Figure. A.2



Author's Contact: mishrasknehu@hotmail.com Ph: 913642550102 (R), 913642723175 (O)