

**Proceedings of the
International Conference on
Mathematics, Geometry,
Statistics, and Computation
(IC-MaGeStiC 2021)**

ISBN 978-94-6239-529-9

ISSN 2352-538X

Advances in
**Computer
Science
Research**

Proceedings of the International Conference on Mathematics, Geometry, Statistics, and Computation (IC-MaGeStiC 2021)

Honorary Chair

Kusbudiono

Universitas Jember, Indonesia

Conference General Chair

Kiswara Agung Santoso

Universitas Jember, Indonesia

Steering Committee

I Made Tirta

Universitas Jember, Indonesia

Moh. Hasan

Universitas Jember, Indonesia

Inne Singgih

University of Cincinnati, United States

Natanael Karjanto

Sungkyunkwan University, South Korea

Novi Herawati Bong

University of Delaware, United States

Technical Committee

Abduh Riski

Universitas Jember, Indonesia

Ikhsanul Halikin

Universitas Jember, Indonesia

Bagus Juliyanto

Universitas Jember, Indonesia

Millatuz Zahroh

Universitas Jember, Indonesia

Organizing Committee

Kristiana Wijaya

Universitas Jember, Indonesia

Yuliani Setia Dewi

Universitas Jember, Indonesia

Alfian Futuhul Hadi

Universitas Jember, Indonesia

Mohamat Fatekurohman

Universitas Jember, Indonesia

Agustina Pradjaningsih

Universitas Jember, Indonesia

Firdaus Ubaidillah

Universitas Jember, Indonesia

Kosala Dwidja Purnomo

Universitas Jember, Indonesia

Treasurer

Ahmad Kamsyakawuni

Universitas Jember, Indonesia

Dian Anggraeni

Universitas Jember, Indonesia

Table of Contents

1. On Ramsey Minimal Graphs for a 3-Matching Versus a Path on Five Vertices

Kristiana Wijaya, Edy Tri Baskoro, Asep Iqbal Taufik, Denny Riama Silaban

2. Ramsey Graphs for a Star on Three Vertices Versus a Cycle

Maya Nabila, Edy Tri Baskoro, Hilda Assiyatun

3. On Ramsey (mK₂, P₄)-Minimal Graphs

Asep Iqbal Taufik, Denny Riama Silaban, Kristiana Wijaya

4. Spectrum of Unicyclic Graph

Budi Rahadjeng, Dwi Nur Yuniarti, Raden Sulaiman, Agung Lukito

5. Distinguishing Number of the Generalized Theta Graph

Andi Pujo Rahadi, Edy Tri Baskoro, Suhadi Widodo Saputro

6. Edge Magic Total Labeling of (n, t)-Kites

Inne Singgih

7. Further Result of H-Supermagic Labeling for Comb Product of Graphs

Ganesha Lapenangga P., Aryanto, Meksianis Z. Ndi

8. Labelling of Generalized Friendship, Windmill, and Torch Graphs with a Condition at Distance Two

Ikhsanul Halikin, Hafif Komarullah

9. On the Minimum Span of Cone, Tadpole, and Barbell Graphs

Hafif Komarullah, Ikhsanul Halikin, Kiswara Agung Santoso

10. L(2,1) Labeling of Lollipop and Pendulum Graphs

Kusbudiono, Irham Af'idatul Umam, Ikhsanul Halikin, Mohamat Fatekurohman

11. Magic and Antimagic Decomposition of Amalgamation of Cycles

Sigit Pancahayani, Annisa Rahmita Soemarsono, Dieky Adzkiya, Musyarofah

12. A Minimum Coprime Number for Amalgamation of Wheel

Hafif Komarullah, Slammin, Kristiana Wijaya

13. Rainbow Connection Number of Shackle Graphs

M. Ali Hasan, Risma Yulina Wulandari, A.N.M. Salman

14. Local Antimagic Vertex Coloring of Corona Product Graphs $P_n \circ P_k$

Setiawan, Kiki Ariyanti Sugeng

15. Local Antimagic Vertex Coloring of Gear Graph

Masdaria Natalina Br Silitonga, Kiki Ariyanti Sugeng

16. Implementations of Dijkstra Algorithm for Searching the Shortest Route of Ojek Online and a Fuzzy Inference System for Setting the Fare Based on Distance and Difficulty of Terrain (Case Study: in Semarang City, Indonesia)

Vani Natali Christie Sebayang, Isnaini Rosyida

17. Granular Dynamic Simulations of Depositing Materials: Two-Dimensional Approach

Mohamad Hasan

18. Active Participation and Student Journal in Confucian Heritage Culture Mathematics Classrooms

Natanael Karjanto

19. Modification Interior-Point Method for Solving Interval Linear Programming

Agustina Pradjaningsih, Fatmawati, Herry Suprajitno

20. Generalization of Chaos Game on Polygon

Kosala D. Purnomo

21. High Order Three-Steps Newton Raphson-like Schemes for Solving Nonlinear Equation Systems

Rizki Multazamil Fatahillah, M Ziaul Arif, Rusli Hidayat, Kusbudiono, Ikhsanul Halikin

22. Symmetry Functions with Respect to Any Point in R_n and Their Properties

Firdaus Ubaidillah

23. Root Water Uptake Process for Different Types of Soil in Unsteady Infiltration from Periodic Trapezoidal Channels

Millatuz Zahroh, Imam Solekhuudin

24. Contact Tracking with Social Network Analysis Graph

Alvida Mustika Rukmi, Wildan Zakky, M. Lutfhi Shahab

25. Learning Materials Development of Parametric Curves and Surfaces for Modeling the Objects Using Maple on Differential Geometry Courses

Kusno, Abduh Riski

26. Hanging Roter Model by Joining Deformation Result of Space Geometry Objects

Een Ubaningrum, Bagus Juliyanto, Ahmad Kamsyakawuni, Firdaus Ubaidillah

27. Learning Content Development in Modeling Creative Industry Objects Using Real Function Formulas Supported with Maple

Kusno, Bagus Juliyanto, Kiswara Agung Santoso

28. SHINY OFFICE-R: A Web-based Data Mining Tool for Exploring and Visualizing Company Profiles

I Made Tirta, Mohamad Fatekurahman, Khairul Anam, Bayu Taruna Widjaja Putra

29. Bayesian Accelerated Failure Time Model and Its Application to Preeclampsia

Dennis Alexander, Sarini Abdullah

30. Correlation Analysis Between the Number of Confirmed Cases of COVID-19 and Stock Trading in Indonesia

Dinagusti Sianturi, Alvida Rukmi

31. Classification of Bank Deposit Using Naïve Bayes Classifier (NBC) and K-Nearest Neighbor (K-NN)

Muhammad Hafidh Effendy, Dian Anggraeni, Yuliani Setia Dewi, Alfian Futuhul Hadi

32. Information Retrieval Using Matrix Methods (Case Study: Three Popular Online News Sites in Indonesia)

Ferry Wiranto, I Made Tirta

33. Statistical Downscaling Technique Using Response Based Unit Segmentation-Partial Least Square (REBUS-PLS) for Monthly Rainfall Forecasting

Izdihar Salsabila, Alfian Futuhul Hadi, I Made Tirta, Yuliani Setia Dewi, Firdaus Ubaidillah, Dian Anggraeni

34. Hurdle Regression Modelling on the Number of Deaths from Chronic Filariasis Cases in Indonesia

Nur Kamilah Sa'diyah, Ani Budi Astuti, Maria Bernadetha T. Mitakda

35. Multiple Discriminant Analysis Altman Z-Score, Multiple Discriminant Analysis Stepwise and K-Means Cluster for Classification of Financial Distress Status in Manufacturing Companies Listed on the Indonesia Stock Exchange in 2019

Hazrina Ishmah, Solimun, Maria Bernadetha Theresia Mitakda

36. Naive Bayes Classifier (NBC) for Forecasting Rainfall in Banyuwangi District Using Projection Pursuit Regression (PPR) Method

Ana Ulul Azmi, Alfian Futuhul Hadi, Yuliani Setia Dewi, I Made Tirta, Firdaus Ubaidillah, Dian Anggraeni

37. Analysis of Factors Affecting Poverty Depth Index in Papua Province Using Panel Data Regression

Rufina Indriani, Erma Oktania Permatasari

38. Projection Pursuit Regression on Statistical Downscaling Using Artificial Neural Network and Support Vector Regression for Rainfall Forecasting in Jember

Chandrika Desyana Putri, Ema Fahma Farikha, Alfian Futuhul Hadi, Yuliani Setia Dewi, I Made Tirta, Firdaus Ubaidillah, Dian Anggraeni

39. Weather Forecasting at BMKG Office Lumajang City Using Markov Chain Method

Ummi Masrurotul Jannah, Mohamat Fatekurohman

40. DOPE: MDC-2 Scheme Using PRESENT

Anjeli Lutfiani, Bety Hayat Susanti

41. Primary Key Encryption Using Hill Cipher Chain (Case Study: STIE Mandala PMB Site)

Muhamat Abdul Rohim, Kiswara Agung Santoso, Alfian Futuhul Hadi

42. A Modification of ECDSA to Avoid the Rho Method Attack

Amira Zahra, Kiki Ariyanti Sugeng

43. Image Authentication Based on Magic Square

Kiswara Agung Santoso, Maulidyah Lailatun Najah, Moh. Hasan

44. Pattern Recognition of Batik Madura Using Backpropagation Algorithm

Abduh Riski, Ega Bandawa Winata, Ahmad Kamsyakawuni

45. Hybrid Cat-Particle Swarm Optimization Algorithm on Bounded Knapsack Problem with Multiple Constraints

Kiswara Agung Santoso, Muhammad Bagus Kurniawan, Ahmad Kamsyakawuni, Abduh Riski

Learning Materials Development of Parametric Curves and Surfaces for Modeling the Objects Using Maple on Differential Geometry Courses

Kusno*, Abduh Riski

Mathematics Department, Faculty of Mathematics and Sciences, University of Jember, Indonesia

*Corresponding author. Email: kusno.fmipa@unej.ac.id

ABSTRACT

Modeling industrial objects needs the formulas of curves and surfaces to construct a precise shape of real goods and simulate some process of form creations. For this reason, the study of the equations of curves and surfaces for objects modeling is essential for resulting in a required shape and feature of the goods. This study aims to enhance the instructional materials of differential geometry for fourth-semester college students. Learning materials help students to be able to design a real object using some parametric formulas of curves and surfaces with the software Maple. The method of research is as follows. (a) Instructional materials design for constructing objects; (b) Formulations and evaluations of graphs for goods modeling; (c) Modeling and simulating to realize these items. The research found some instructional materials and parametric formulas of curves and surfaces to equip students to design and evaluate real objects and home industry goods. The use of Maple can help them to present the graphs and the simulation process. The contributions of this study support the students to learn autonomously and creatively with their knowledge, technological skills, and experiences in implementing some differential geometry formulas (especially the curves and surfaces) for designing objects using the Maple tool.

Keywords: *Development, Learning materials, Curves and surfaces, Modeling, Parametric formulas, Maple.*

1. INTRODUCTION

Modeling industrial objects and some mechanical components can use curves and surfaces equations in the form of parametric representations [1,2]. Developing computer-aided design and manufacturing technologies needs the curves and surfaces formula to construct a precise shape of a real object and simulate some process of form creations. For example, using the curves can create many models of pipes [3-5], and the surfaces formulas can apply to design the shape of car bodies, electronic equipment, or cottage industry objects. On the other hand, we can implement the developable surfaces to draw surfaces of big objects constructed by plywood or plat-metal sheets, for example, aircraft, ship hulls industries, and trains [6-8]. For the reasons, the students learning formulation of curves and surfaces with software skills are vital to understanding the creating surfaces forms of the designed industrial objects [9-11]. It is to provide practical experience for the learners in modeling various industries' object shapes.

Object modeling is an activity to select, describe, and create a shape to meet a required function. This shape selection means the geometric modeling in the design process included graphics representations and analytic formulas of curves and surfaces. For this reason, it needs some equations of parametric curves and surfaces in the instructional materials of differential geometry that can use to create many different forms of surfaces of objects. The use of these formulas can produce an aesthetic shape of the designed goods. This paper discusses the development's learning materials for modeling the real goods and cottage industry objects using parametric representations supported with software-Maple.

This study aims to enhance the instructional materials of differential geometry for fourth-semester college students. Learning materials help students to be able to design a real object using some parametric formulas of curves and surfaces with the software Maple. This study contributes to students learning autonomously and creatively with their knowledge, technological skills, and

experiences in implementing some differential geometry formulas (especially the curves and surfaces) for designing objects using the Maple tool.

2. METHOD

The method of research uses development research. This study deal with the understanding of learning, the design of learning contents, and the development of tools. The research results may find models, constructs, or instantiations that can significantly enhance the understanding of learning subjects matters [12,13]. In this case, we identify and explain the process of object design and development of teaching materials of the differential geometry courses using Maple [14,15]. The steps of research are as follows [16]. (a) Instructional materials design for constructing objects; (b) formulations and evaluations of graphs for goods modeling; (c) Modeling and simulating the industrial goods. The use of Maple is to support the students to obtain some design experiences during a simulation. Additionally, this software Maple may help the learners to immediate feedback to alter the model, re-test it quickly, and evaluate their modeling processes and results.

3. RESULTS AND DISCUSSION

3.1. Instructional Materials Design for Constructing Objects

We can present the parametric functions of curves and surfaces in 3D respectively in form $C(u): \mathcal{R}^1 \rightarrow \mathcal{R}^3$ and $S(u,v): \mathcal{R}^2 \rightarrow \mathcal{R}^3$ and the algebraic representation of these curves and surfaces are

$$C(u) = \langle x(u), y(u), z(u) \rangle = x(u) \mathbf{i} + y(u) \mathbf{j} + z(u) \mathbf{k} \quad (1a)$$

$$S(u,v) = \langle x(u,v), y(u,v), z(u,v) \rangle = x(u,v) \mathbf{i} + y(u,v) \mathbf{j} + z(u,v) \mathbf{k} \quad (1b)$$

with $x(u)$, $y(u)$, $z(u)$, $x(u,v)$, $y(u,v)$, and $z(u,v)$ the functions of real values. Developing instructional materials of the curves and the surfaces suitable with students' knowledge, learners skill in using Maple, and modeling objects form needs the investigation and the evaluation to the parametric formulas. Based on the instructional objectives of the differential geometry course, this research discusses to equip the students in designing the shapes of some industry objects. In this case, it evaluates some curves and surfaces formulas for constructing handicrafts and pipes shape.

3.1.1. Some Parametric Presentations of Curves

Some curves in the parametric form are presented as follows.

1. The formula of line segment \overline{PQ} with the endpoints P and Q of the vectors position $\mathbf{P} \langle x_p, y_p, z_p \rangle$ and $\mathbf{Q} \langle x_q, y_q, z_q \rangle$ is

$$L(u) = \langle (1-u) \cdot x_p + u \cdot x_q, (1-u) \cdot y_p + u \cdot y_q, (1-u) \cdot z_p + u \cdot z_q \rangle \quad (2)$$

with $0 \leq u \leq 1$.

2. Consider two constant unity vectors perpendicular in space $\mathbf{u}_1 = \langle a_1, a_2, a_3 \rangle$ and $\mathbf{u}_2 = \langle b_1, b_2, b_3 \rangle$. It can define an ellipse laid in the plane $[\mathbf{u}_1, \mathbf{u}_2]$ of the center $\langle x_c, y_c, z_c \rangle$ in the expression

$$\mathbf{E}(u) = \langle x_c, y_c, z_c \rangle + [p \cdot \cos \varphi \mathbf{u}_1 + q \cdot \sin \varphi \mathbf{u}_2] \quad (3)$$

with p and q positive real values, $\varphi = 2\pi u$ and $0 \leq u \leq 1$. If the curves are a circle, a limaçon, and a rose, then it can define the parametric formulas respectively [3]

$$C_1(u) = r_1(u) \cdot [\cos \varphi \mathbf{u}_1 + \sin \varphi \mathbf{u}_2] \quad (4a)$$

$$C_2(u) = r_2(u) \cdot [\cos \varphi \mathbf{u}_1 + \sin \varphi \mathbf{u}_2] \quad (4b)$$

$$C_3(u) = r_3(u) \cdot [\cos \varphi \mathbf{u}_1 + \sin \varphi \mathbf{u}_2] \quad (4c)$$

with the real functions $r_1(u) = p$; $r_2(u) = p \pm q \cdot \sin \varphi$ or $r_2(u) = p \pm q \cdot \cos \varphi$; $r_3(u) = p \pm q \cdot \sin(n \cdot \varphi)$ or $r_3(u) = p \pm q \cdot \cos(n \cdot \varphi)$. In this case, the values p and q are positive real constants, $\varphi = 2\pi \cdot u$ and $0 \leq u \leq 1$, with the number of rose leaves n .

3. The heliks with the center (x_1, y_1, z_1) is

$$\mathbf{H}_L(u) = \langle x_1 + p \cdot \cos u, y_1 + q \cdot \sin u, z_1 + u \rangle. \quad (5)$$

4. Let a regular curve $C(u)$ of Equation (1a) differentiated twice in interval $0 \leq u \leq 1$. In the natural parameter s , the tangent unit vector \mathbf{t} , the normal unit vector \mathbf{n} , and the unit binormal vector \mathbf{b} of the curve $C(u)$ are orthogonal to each other in the form [5]

$$\mathbf{t} = \frac{dC(u)}{ds} = \dot{C} = \frac{C'}{|C'|} \quad (6a)$$

$$\mathbf{n} = -\frac{\mathbf{k}}{|\mathbf{k}|} \text{ and } \mathbf{k} = \frac{d\mathbf{t}}{ds} = \dot{\mathbf{t}} \quad (6b)$$

$$\mathbf{b} = \mathbf{t} \wedge \mathbf{n}. \quad (6c)$$

3.1.2. Some Parametric Presentations of Surfaces

Some surfaces in the parametric form are presented as follows.

1. The parametric formula of sphere or ellipsoid of cente $\langle x_c, y_c, z_c \rangle$ is successively

$$S_p(u,v) = \langle r \cdot \sin u \cos v + x_c, r \cdot \sin u \cdot \sin v + y_c, r \cdot \cos u + z_c \rangle \quad (7a)$$

$$\mathbf{E}(u,v) = \langle r_1 \cdot \sin u \cos v + x_c, r_2 \cdot \sin u \cdot \sin v + y_c, r_3 \cdot \cos u + z_c \rangle \quad (7b)$$

with r , r_1 , r_2 , and r_3 positive real constants.

2. The cylinder of radius r with the axe (x_c, y_c, z_c) is

$$\mathbf{C}_r(u,v) = \langle r \cdot \cos u + x_c \cdot v, r \cdot \sin u + y_c \cdot v, z_c \cdot v \rangle$$

(8)

with r positive real.

3. The hyperboloid of center $\langle x_c, y_c, z_c \rangle$ is

$$\mathbf{H}(u,v) = \langle (\cos u - v \cdot \sin u) + x_c, (\sin u + v \cdot \cos u) + y_c, v + z_c \rangle$$

(9)

4. The paraboloid of radius r and center $\langle x_c, y_c, z_c \rangle$ is

$$\mathbf{P}(u,v) = \langle r \cdot v \cdot \cos u + x_c, r \cdot v \cdot \sin u + y_c, r_1 \cdot v^2 + z_c \rangle$$

(10)

with r, r_1 positive real constants.

5. The cone of radius r and center $\langle x_c, y_c, z_c \rangle$ is

$$\mathbf{K}(u,v) = \langle r \cdot v \cdot \cos u + x_c, r \cdot v \cdot \sin u + y_c, r_1 \cdot v + z_c \rangle$$

(11)

with r, r_1 positive real constants.

6. The circle-plane of radius r , height z_1 , and center $\langle x_c, y_c, z_c \rangle$ is

$$\mathbf{C}_L(u,v) = \langle r \cdot u \cdot \cos v + x_c, r \cdot u \cdot \sin v + y_c, z_c \rangle$$

(12)

7. The pseudo-sphere of radius r is

$$\mathbf{P}_B(u,v) = \langle r/v \cdot \cos u, r/v \cdot \sin u, r_1 \cdot v \rangle$$

(13)

with r, r_1 positive real constants.

3.2. Formulations and Evaluations of Graphs for Objects Modeling

Helping students plot the graphs and implement Equation 1 up to Equation 15 to design real objects is necessary for the learners to evaluate these curves and surfaces formula using software-Maple [11,17,18]. Then, they try to create more complex graphs by connecting some curves and surfaces for modeling the object shapes. This section will study Equation 1-15 deals with the presentation aspects of the curves and surfaces, and the formulation method with Maple.

Line segment formula of Equation 2 can directly plot the line segment \overline{PQ} from the vector positions $\mathbf{P} \langle x_p, y_p, z_p \rangle$ to the vector position $\mathbf{Q} \langle x_q, y_q, z_q \rangle$ without tangent line calculation and determination of parameter value u . For example, when the vector position of the endpoint points P and Q are respectively $\mathbf{P} \langle 5, 2, 1 \rangle$ and $\mathbf{Q} \langle 0, 8, 6 \rangle$, it finds Figure 1a. Then, using Formula 3 and 4 can define an ellips, a circle, a limaçon, and a rose that are laid in the plane of unity vectors $\mathbf{u}_1 = \langle a_1, a_2, a_3 \rangle$ and $\mathbf{u}_2 = \langle b_1, b_2, b_3 \rangle$. The plotted shapes, generally, are symmetric and closed curves. These curves type usually use to construct the shapes of flower vases' rack, plane decorations, and model various forms of pipes cross-section. If we pose $\mathbf{u}_1 = \langle \langle 3/5, 4/5, 0 \rangle \rangle$, $\mathbf{u}_2 = \langle 0, 0, 1 \rangle$ and the Equation (3) is in the form $\mathbf{E}(u) = \langle 2, 2, 2 \rangle + [5$

$\cos \varphi \mathbf{u}_1 + 8 \cdot \sin \varphi \mathbf{u}_2]$, then it gets Figure 1b. When Equations 4a, 4b, and 4c are $\mathbf{C}_1(u) = 6 \cdot [\cos \varphi \mathbf{u}_1 + \sin \varphi \mathbf{u}_2]$, $\mathbf{C}_2(u) = (2+4 \cdot \cos \varphi) \cdot [\cos \varphi \mathbf{u}_1 + \sin \varphi \mathbf{u}_2]$, and $\mathbf{C}_3(u) = (3+\cos(3 \cdot \varphi)) \cdot [\cos \varphi \mathbf{u}_1 + \sin \varphi \mathbf{u}_2]$, it can obtain Figure 1c, 1d, and 1e, respectively. In case the helix formulation of Equation 5 in the form $\mathbf{H}_L(u) = \langle 2 + 4 \cdot \cos u, 1 + 6 \cdot \sin u, 3 + u \rangle$ with $0 \leq u \leq 4\pi$, it will give Figure 1f. On the other hand, consider a curve $\mathbf{C}(u) = \langle 17u^3 - 30u^2 + 6u - 12, -39u^3 + 51u^2 - 36u + 10, 203u^3 + 393u^2 - 195u + 15 \rangle$ and the parameter u in interval $0 \leq u \leq 1$. We can draw the curve $\mathbf{C}(u)$ in the form of Figure 1g and calculate the tangent vectors \mathbf{t} of the curve $\mathbf{C}(u)$ at $u = 0$ and $u = 1$ as shown in Figure 1h. These defined parametric curves in Figure 1a-1h are presented using Maple with the syntaxes as shown in Table 1.

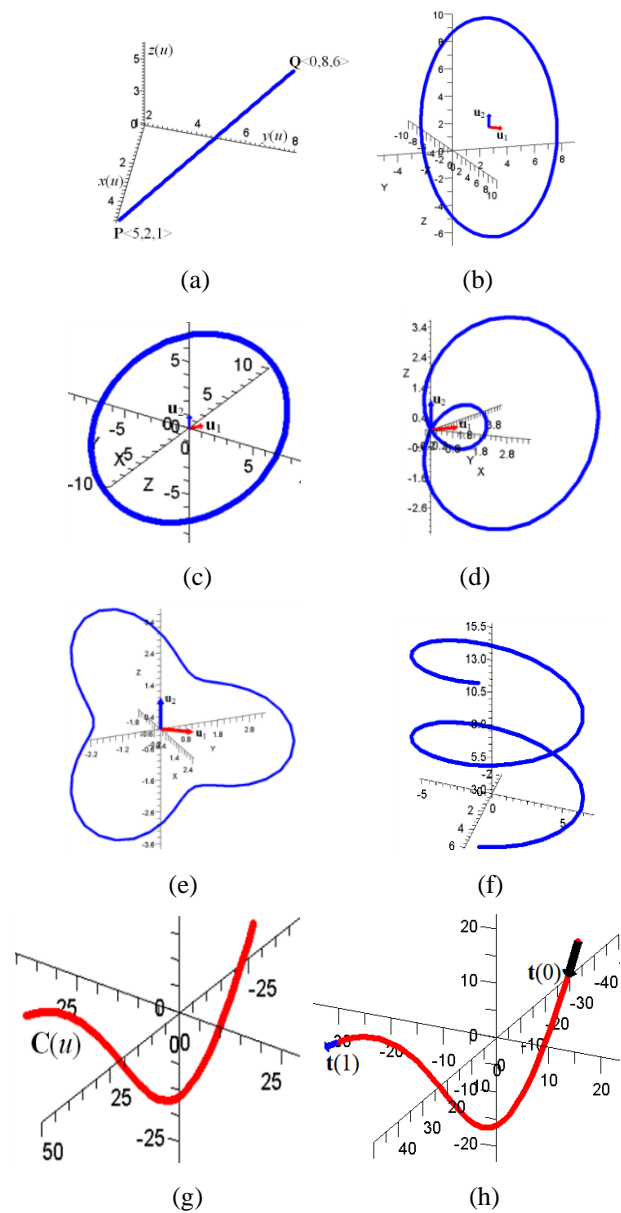


Figure 1 Some parametric curves.

Table 1. Plotting the parametric curves using Maple

No.	Equations Representation of Figures 1a-1h Using Maple	Explanations
	<code>> with(plots):</code>	
1.	<code>> spacecurve([(1-u)*5+u*0, (1-u)*2+8*u, 1-u+6*u], u = 0 .. 1, thickness = 3, color = blue);</code> <code>>a[1] := spacecurve([2+5*cos(2*Pi*u)*3/5+8*sin(2*Pi*u)*0, 2+5*cos(2*Pi*u)*4/5+8*sin(2*Pi*u)*0, 2+5*cos(2*Pi*u)*0+8*sin(2*Pi*u)*1], u = 0 .. 1):</code>	Figure 1a
2.	<code>>u1 := arrow(<,>(2, 2, 2), <,>(3/5, 4/5, 0), width = [.2, relative], color = red):</code> <code>>u2 := arrow(<,>(2, 2, 2), <,>(0, 0, 1), width = [.2, relative], color = blue):</code> <code>display(a[1], u1, u2, color = blue, thickness = 3, labels = [X, Y, Z]);</code>	Figure 1b
3.	<code>>a[2] := spacecurve([7*(cos(2*Pi*u)*3/5+sin(2*Pi*u)*0), 7*(cos(2*Pi*u)*4/5+sin(2*Pi*u)*0), 7*(cos(2*Pi*u)*0+sin(2*Pi*u)*1)], u = 0 .. 1):</code>	Figure 1c
4.	<code>>a[3] := spacecurve([(2+4*cos(2*Pi*u))*(cos(2*Pi*u)*3/5+sin(2*Pi*u)*0), (2+4*cos(2*Pi*u))*(cos(2*Pi*u)*4/5+sin(2*Pi*u)*0), (2+4*cos(2*Pi*u))*(cos(2*Pi*u)*0+sin(2*Pi*u)*1)], u = 0 .. 1):</code>	Figure 1d
5.	<code>>a[4] := spacecurve([(3+cos((3*2)*Pi*u))*(cos(2*Pi*u)*3/5+sin(2*Pi*u)*0), (3+cos((3*2)*Pi*u))*(cos(2*Pi*u)*4/5+sin(2*Pi*u)*0), (3+cos((3*2)*Pi*u))*(cos(2*Pi*u)*0+sin(2*Pi*u)*1)], u = 0 .. 1);</code>	Figure 1e
6.	<code>> spacecurve([2+4*cos(u), 1+6*sin(u), 3+u], u = 0 .. 4*Pi, thickness = 3, color = blue);</code>	Figure 1f
7.	<code>> spacecurve([(17*u*u)*u-30*u*u+60*u-12, -(39*u*u)*u+51*u*u-36*u+10, -(203*u*u)*u+393*u*u-195*u+15], u = 0 .. 1, thickness = 3, color = red, view = [-50 .. 50, -50 .. 50, -50 .. 50]);</code> <code>>a[5] := spacecurve([(17*u*u)*u-30*u*u+60*u-12, -(39*u*u)*u+51*u*u-36*u+10, -(203*u*u)*u+393*u*u-195*u+15], u = 0 .. 1, thickness = 3, color = red):</code>	Figure 1g
8.	<code>>t(0) := arrow(<,>(-12, 10, 15), <,>(2, -1.2, -6.5), width = [0.2, relative], color = black):</code> <code>>t(1) := arrow(<,>(35, -14, 10), <,>(1.7, -1.7, -6), width = [0.4, relative], color = blue):</code> <code>display(a[5], t(0), t(1), view = [-50 .. 50, -50 .. 50, -50 .. 50]);</code>	Figure 1h

From Equation 7 to the Equation 13, it can demonstrate the parametric surfaces in this ways. The sphere and ellipsoid in Equation 7 of center $\langle 3, 2, 4 \rangle$ with the radius $r = 6$, $r_1 = 3$, $r_2 = 5$, $r_3 = 8$, and $0 \leq u \leq 1.8\pi$, and $0 \leq v \leq \pi$ produce the shapes in Figure 2a and 2b. Then, the cylinder of Equation 8 with the radius $r = 5$, the axe $(0.5, 2, 2)$, $0 \leq u \leq 2\pi$, and $0 \leq v \leq 12$ presents Figure 2c. Formulation of hyperboloid, paraboloid, cone, and circle-plane of Equation 9, 10, 11, and 12 in the forms $\mathbf{H}(u, v) = \langle (2 \cdot \cos u - 2 \cdot v \cdot \sin u) + 2, (2 \cdot \sin u + 2 \cdot v \cdot \cos u) + 2, 3 \cdot v + 3 \rangle$ with $0 \leq u \leq 2\pi$ and $-3 \leq v \leq 3$, $\mathbf{P}(u, v) = \langle 3 \cdot v \cdot \cos u + 1, 3 \cdot v \cdot \sin u + 2, 2 \cdot v^2 + 3 \rangle$ with $0 \leq u \leq 2\pi$ and $-3 \leq v \leq 3$, $\mathbf{K}(u, v) = \langle 5 \cdot v \cdot \cos u + 2, 5 \cdot v \cdot \sin u + 3, 7 \cdot v + 1 \rangle$ with $0 \leq u \leq 2\pi$ and $0 \leq v \leq 5$, and $\mathbf{C}_L(u, v) = \langle 6 \cdot v \cdot \cos u + 1, 6 \cdot v \cdot \sin u + 3, 4 \rangle$ with $0 \leq u \leq 2\pi$, and $0 \leq v \leq 1$ give the surfaces presented in Figure 2d, 2e, 2f, and 2g, successively. Finally, the pseudo-sphere of formula $\mathbf{P}_B(u, v) = \langle 12/v \cdot \cos u, 12/v \cdot \sin u, 5 \cdot v \rangle$ with $-\pi \leq u \leq \pi$ and $-1 \leq v \leq -5$ presents the Figure 2h.

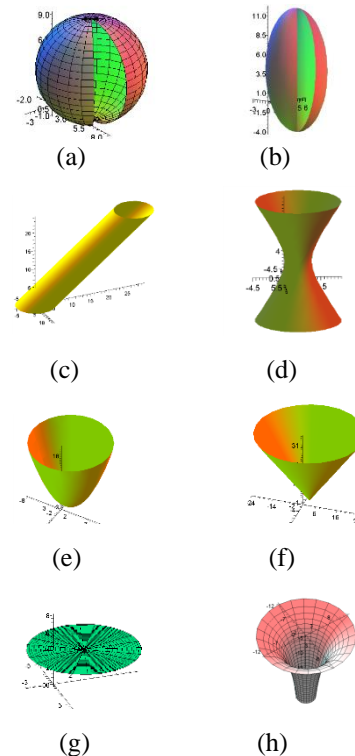


Figure 2 Some parametric surfaces.

Table 2. Plotting the parametric surfaces using Maple

No.	Equations Representation of Figures 2a-2h Using Maple	Explanations
1.	<code>> with(plots): > plot3d([5*sin(v)*cos(u)+3, 5*sin(v)*sin(u)+2, 5*cos(v)+4], u = 0 .. 1.8*Pi, v = 0 .. Pi):</code>	Figure 2a
2.	<code>> plot3d([3*sin(v)*cos(u)+3, 5*sin(v)*sin(u)+2, 8*cos(v)+4], u = 0 .. 1.8*Pi, v = 0 .. Pi):</code>	Figure 2b
3.	<code>> plot3d([5*cos(u)+0.5*v, 5*sin(u)+2*v, 2*v], u = 0 .. 2*Pi, v = 0 .. 12):</code>	Figure 2c
4.	<code>> plot3d([2*cos(u)-2*v*sin(u)+2, 2*sin(u)+2*v*cos(u)+2, 3*v+3], u = 0 .. 2*Pi, v = -3 .. 3):</code>	Figure 2d
5.	<code>> plot3d([3*v*cos(u)+1, 3*v*sin(u)+2, 2*v*v+3], u = 0 .. 2*Pi, v = -3 .. 3):</code>	Figure 2e
6.	<code>> plot3d([5*v*cos(u)+2, 5*v*sin(u)+3, 7*v+1], u = 0 .. 2*Pi, v = 0 .. 5):</code>	Figure 2f
7.	<code>> plot3d([6*v*cos(u)+1, 6*v*sin(u)+3, 4], u = 0 .. 2*Pi, v = 0 .. 1):</code>	Figure 2g
8.	<code>> plot3d([(12/v)*cos(u), (12/v)*sin(u), 5*v], u = -Pi .. Pi, v = -1 .. -5):</code>	Figure 2h

Consider two parametric curves of Equation 1a in the forms $C_1(u)$ and $C_2(u)$ with $0 \leq u \leq 1$. The connection of both curves are respectively a parametric continuity order 0 and 1, i.e., C^0 and C^1 , if they meet $C_1(1) = C_2(0) = C_P$ and $C'_1(1) = C'_2(0)$. If the condition of tangent vectors $C'_1(1)$ and $C'_2(0)$ are collinear, it fulfills the geometric continuity. In case the surfaces $S_1(u,v)$ and $S_2(u,v)$ with $0 \leq u,v \leq 1$, they have to fulfill $S_1(1,v) = S_2(0,v) = \rho(v)$ and $S_1^u(1,v) = S_2^u(0,v)$ along the common curve $\rho(v)$.

Connection of these parametric curves and surfaces can use to model the shape of an object. So, it needs the calculation of common boundary point C_P and the computation of collinear tangent vectors $C'_1(1) = C'_2(0)$ of the curves joints. On the other sides, we have to evaluate the common boundary curve $\rho(v)$ and the derivation in direction u of the surfaces $S_1^u(1,v) = S_2^u(0,v)$ or the unique tangent plane along the common curve $\rho(v)$ of the surfaces joints. For examples, let $C_1(u) = \langle 2u^2, 2u-1, 3u-2 \rangle$, $C_2(u) = \langle 2u^2+4u+2, 2u^3+2u+1, 3u+1 \rangle$ and $0 \leq u \leq 1$. We can calculate the common boundary point $C_1(1) = C_2(0) = \langle 2, 1, 1 \rangle$, and the common tangent vector $C'_1(1) = C'_2(0) = \langle 4, 2, 3 \rangle$ as shown in Figure 3a and 3b.

When the surfaces $S_1(u,v)$ and $S_2(u,v)$ are $S_1(u,v) = \langle (2u^2+2).cos(2\pi v), (2u^2+2).sin(2\pi v), 4u+1 \rangle$ and $S_2(u,v) = \langle (2u^2+4u+4).cos(2\pi v), (2u^2+4u+4).sin(2\pi v), 2u^3+u^2+4u+5 \rangle$ with $0 \leq u,v \leq 1$, we find $S_1(1,v) = S_2(0,v) = \rho(v) = \langle 4.cos(2\pi v), 4.sin(2\pi v), 5 \rangle$ as shown in Figure 3c. The derivation in direction u of both surfaces along the common boundary curve $\rho(v)$ is $S_1^u(1,v) = S_2^u(0,v) = \langle 4.cos(2\pi v), 4.sin(2\pi v), 4 \rangle$. The connection of these surfaces is presented in Figure 3d.

The parametric curves and surfaces formulas of Equation 2-12 are demonstrated in Table 1 and Table 2 using Maple. These formulas may give some advantages to model various forms of home industry objects. Then, joining the curves and surfaces may also be used to construct the object components and the required goods.

Based on these studies, we discuss, in the next section, the construction of the cottage industry and pipes.

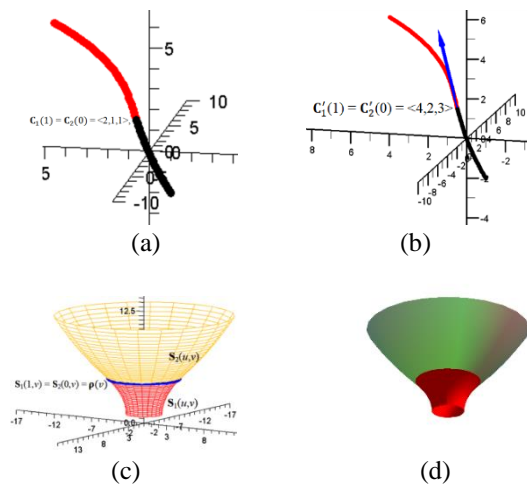


Figure 3 Connection of parametric curves and surfaces.

3.3. Modeling and Simulating Industrial Objects

3.3.1. Design of Home Industrial Objects

We show the use of these curves for creating iron trellis and frames, for example, to construct the flower vases' rack. Making the designed rack be symmetric and balanced form, it can reference from a cube or rectangular parallelepiped shapes $ABCD-EFGH$ as follows. From Figure 4a, we determine the line segments $KA=DK=KL$, $PB=PC=PO$, and $LM=MN=NO$. In the plane $KLOP$, it alternately draws the semicircles with diameter LM , MN , and NO . In-plane $AEHD$ and $BFGC$, it can plot the semicircles of radius KA and PB with the center K and P , respectively, as presented in Figure 4b. Then, it can pose some circles with centers at the points L, M, N, O , and the summit points of the semicircles to provide the places of flower vases. The design result may introduce in Figure 4c, 4d.

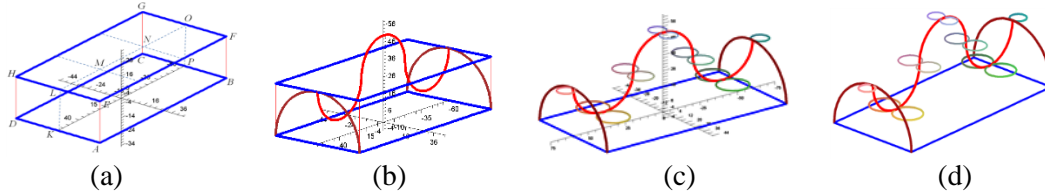


Figure 4 Rack design (flower vase).

Table 3. Plotting rack design using curves (Figure 4c,4d)

Curves Formulas Using Maple of Figure
<pre> > with(plots): p1 := spacecurve({[(1-u)*60+u*(-60), (1-u)*(-30)+u*(-30), (1-u)*0+u*0], [(1-u)*60+u*(-60), (1-u)*30+30*u, (1-u)*0+u*0]}, u = 0 .. 1, thickness = 4, color = blue): p2 := spacecurve({[(1-u)*(-60)+u*(-60), (1-u)*30+u*(-30), (1-u)*0+u*0], [(1-u)*60+60*u, (1-u)*30+u*(-30), (1-u)*0+u*0]}, u = 0 .. 1, thickness = 4, color = blue): t1:=spacecurve({[(20)*cos(u*2*Pi)+40,0,20*sin(u*2*Pi)+30],[20*cos(u*2*Pi)-40,0,20*sin(u*2*Pi)+30], [(20)*cos(u*2*Pi-Pi)+0,0,20*sin(u*2*Pi-Pi)+30]}, u=0.5..1, thickness=5, color=red): k12:=spacecurve({[-60,(30)*cos(u*Pi),30*sin(u*Pi)], [60,(30)*cos(u*Pi),30*sin(u*Pi)]}, u=0..1, thickness=5, color= brown): C:=spacecurve({[5*cos(u*2*Pi)+65,5*sin(u*2*Pi),30],[5*cos(u*2*Pi)-65,5*sin(u*2*Pi),30]}, u=0..1, thickness = 4): C2:=spacecurve({[9*cos(u*2*Pi)+40,9*sin(u*2*Pi)+9,10],[9*cos(u*2*Pi)+40,9*sin(u*2*Pi)-9,10]}, u=0..1, thickness = 4): C3:=spacecurve({[7*cos(u*2*Pi)+20,7*sin(u*2*Pi)+7,30],[7*cos(u*2*Pi)+20,7*sin(u*2*Pi)-7,30]}, u=0..1, thickness = 3): C4:=spacecurve({[5*cos(u*2*Pi)+0,5*sin(u*2*Pi)+5,50],[5*cos(u*2*Pi)+0,5*sin(u*2*Pi)-5,50]}, u=0..1, thickness = 3): C5:=spacecurve({[7*cos(u*2*Pi)-20,7*sin(u*2*Pi)+7,30],[7*cos(u*2*Pi)-20,7*sin(u*2*Pi)-7,30]}, u=0..1, thickness = 3): C6:=spacecurve({[9*cos(u*2*Pi)-40,9*sin(u*2*Pi)+9,10],[9*cos(u*2*Pi)-40,9*sin(u*2*Pi)-9,10]}, u=0..1, thickness = 4): display(p1,p2,t1,k12,C,C2,C3,C4,C5,C6,view = [-75 .. 75, -45 .. 45, -5 .. 65]); </pre>

Modeling handicraft and home industries objects with surfaces need to recognize various component features, the measure, the position of the parts, and the aesthetic of the goods [19-21]. In this case, Equation 7-13 and some modifications formula provide some facilitates to create the shapes and the model of the goods using stages as follows. First, we determine each position of the part object and its measures from bottom to top, from back to front, or from left to right, consecutively. Second, connecting some surfaces of Equation 7-13 using a Maple suitable with the arranged positions will create the shape of the part of the goods. Finally, showing naturally, the designed objects can use the Maple menu related to the presentation style of surfaces, color, or lighting. For example, implementing the formula cone and paraboloid (Equation 11, 10) ordered from bottom to top can model a hat shape (Figure 5a). In the same manner, it can create a model of Figure 5b and Figure 5c. Moreover, the application of many Equations 7-13 gives the complex shapes as illustrated in Figures 5d, 5e, and 5f. Table 4 shows some equations used to draw Figures 5a, 5b, and 5d using software-Maple.

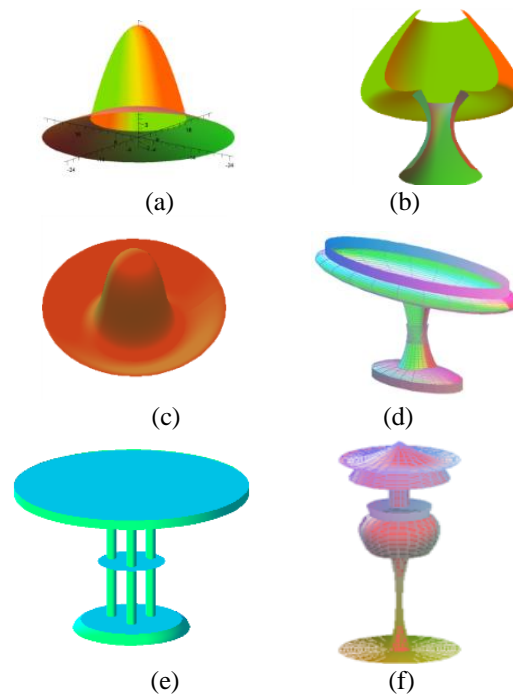


Figure 5 Simulation results of the object design.

Table 4. Objects design constructed with surfaces

Figures	Surfaces Formulas Using Maple
5a	<pre>> with(plots): T[1] := plot3d([(5*(-2*v+4))*cos(2*Pi*u), (5*(-2*v+4))*sin(2*Pi*u), 4*v], u = 0 .. 1, v = 0 .. 1): T[2] := plot3d([10*v*cos(2*Pi*u), 10*v*sin(2*Pi*u), -20*v*v+24], u = 0 .. 1, v = 0 .. 1, color = red): display(T[1], T[2], view = [-25 .. 25, -25 .. 25, -2 .. 30]);</pre>
5b	<pre>> with(plots): p1 := plot3d([(6*v*v-7*v+3) *cos(1.5*Pi*u), (6*v*v-7*v+3) *sin(1.5*Pi*u), -v*v+7*v], u = 0 .. 1, v = 0 .. 1): p2 := plot3d([(-15*v*v+15*v+2) *cos(1.5*Pi*u), (-15*v*v+15*v+2) *sin(1.5*Pi*u), 11*v*v-17*v+ 12], u = 0 .. 1, v = 0 .. 1): display(p1, p2, view = [-7 .. 7, -7 .. 7, 0 .. 12]);</pre>
5d	<pre>> with(plots): T[1] := plot3d([4*sin(v)*cos(u)/(.866), 2*sin(v)*sin(u)/(.866), cos(v)+.25], u = 0 .. 2*Pi, v = 0 .. (1/3)*Pi): T[2] := plot3d([4*cos(u), 2*sin(u), v+.25], u = -Pi .. Pi, v = .5 .. 0): T[3] := plot3d([cos(u)/v, sin(u)/v, -2*v+3.5-4], u = -Pi .. Pi, v = -2 .. -.82): T[4] := plot3d([cos(u)/v, sin(u)/v, 2*v+3+4], u = -Pi .. Pi, v = -2 .. -1): T[5] := plot3d([.6*(cos(u)-v*sin(u)), .6*(sin(u)+v*cos(u)), v+3], u = 0 .. 2*Pi, v = -.5 .. .5): T[6] := plot3d([9*sin(v)*cos(u)/(.9), 3*sin(v)*sin(u)/(.9), cos(v)+6], u = 0 .. 2*Pi, v = Pi .. (1/3)*Pi): T[7] := plot3d([9*cos(u), 3*sin(u), v], u = -Pi .. Pi, v = 6.5 .. 7.5): display(T[1],T[2],T[3],T[4],T[5],T[6],T[7], view = [-12 .. 12, -12 .. 12, -12 .. 12]);</pre>

3.3.2. Pipes Design

Consider two perpendicular unity constant vectors \mathbf{v}_1 and \mathbf{v}_2 with $\mathbf{v}_1 \wedge \mathbf{v}_2 = \mathbf{v}$ and the parametric curve $\mathcal{I}(v)$ in the form of Equation (4). On the other side, consider a curve $\mathbf{C}(u)$ with their normal and binormal vectors \mathbf{n} and \mathbf{b} shown in Figure 6a. It can define a tube with a center of the curve, i.e., a line or the curve $\mathbf{C}(u)$ and the cross-section curve of the pipe $\mathcal{I}(v)$ in this way [3,5]

$$\mathbf{T}_v(u, v) = [\langle x_C, y_C, z_C \rangle + \lambda u, \mathbf{v}] + \mathcal{I}(v) \cdot [\cos(\varphi) \mathbf{v}_1 + \sin(\varphi) \mathbf{v}_2] \quad (14)$$

$$\mathbf{T}(u, v) = \mathbf{C}(u) + \mathcal{I}(v) \cdot [\cos(\varphi) \mathbf{b} + \sin(\varphi) \mathbf{n}] \quad (15)$$

with λ parameter constant, the value $\varphi = 2\pi v$ and $0 \leq u, v \leq 1$. For example, if the unity constant vectors $\mathbf{v}_1 = \langle 0, 0, 1 \rangle$, $\mathbf{v}_2 = \langle 0, 1, 0 \rangle$, $\langle x_C, y_C, z_C \rangle = \langle 5, 0, 0 \rangle$ and $\mathcal{I}(v) = 2 + \cos(6\pi v)$, then we find Figure 6b. When the center curve of pipe $\mathbf{C}(u) = \langle 17u^3 - 30u^2 + 6u - 12, -39u^3 + 51u^2 - 36u + 10, 203u^3 + 393u^2 - 195u + 15 \rangle$ and $\mathcal{I}(v) = 3$, using Equation (6) and Equation (15) will obtain the tube shape in Figure 6c, 6d.

4. CONCLUSIONS

This study develops learning materials to apply the parametric formulas of curves and surfaces to design the shape of cottage industry, pipes shape and modeling the handicrafts with software-Maple. In modeling these goods, software-Maple helps the students recognize and

implement the curves and surfaces formula, visualize and present the industrial objects naturally. Moreover, aiding Maple, the collegers learn creatively, visually, and responsibly with their knowledge, technological skills, and experiences in implementing the differential geometry formulas (especially the curves and surfaces) for modeling industrial objects.

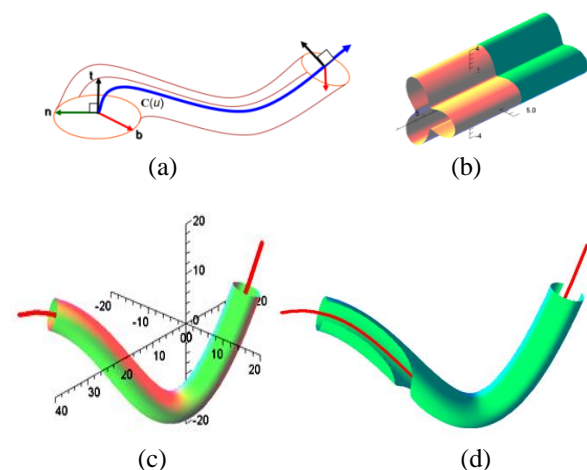


Figure 6 Pipes construction.

REFERENCES

- [1] M.E. Mortenson, *Geometry Modeling*, John Wiley and Sons Inc., New York, 1996.
- [2] J. Oprea, *Differential Geometry and its Applications*, The Mathematical Association of America, Washington DC, 2007.
- [3] Kusno, On the modeling of cross-section and longitudinal section of pipes, *Journal of Physics Conference Series*, vol. 321(2), 2019, pp. 1-8. DOI: <https://doi.org/10.1088/1742-6596/1321/2/022068>
- [4] Kusno, Modeling pipes using pipes' center curves of quadratic and cubic spline interpolation, *AIP Conference Proceedings*, vol. 2329, 2021. DOI: <https://doi.org/10.1063/5.0042248>
- [5] Kusno, A. C. Prihandoko, Continuous connection of two adjacent pipe parts defined by line, Bézier and Hermit center curves, *Journal of Physics Conference Series*, vol. 1008(1), 2018, pp. 1-7. DOI: <https://doi.org/10.1088/1742-6596/1008/1/012005>
- [6] Kusno, Modeling of developable surfaces using Hermite spline interpolation curves, *Advances in Mathematics: Scientific Journal*, vol. 9(10), 2020, pp. 8431–8442. DOI: <https://doi.org/10.37418/amsj.9.10.72>
- [7] Kusno, Fitting a curve, cutting surface, and adjusting the shapes of developable Hermite patches, *Mathematics and Statistics*, vol. 8 (6), 2020, pp. 740-746. DOI: <https://doi.org/10.13189/ms.2020.080615>
- [8] Kusno, On the modeling of developable hermite patches, *Journal of Mathematical and Computational Science*, vol. 11(2), 2021, pp.1145–1165. DOI: <https://doi.org/10.28919/jmcs/5293>
- [9] F. Yamaguchi, *Curves and Surfaces in Computer Aided Geometric Design*, Springer-Verlag, Berlin, 1988.
- [10] S. Montiel, A. Ros, *Curves and Surfaces*, American Mathematical Society, Providence, 2009.
- [11] M. L. Abell, J. P. Braselton, *Maple by Example*, Elsevier Inc, Amsterdam, 2005.
- [12] A-K. Carstensen, J. Bernhard, Design Science Research – an engineering research approach to improve methods for engineering education research, *Research in Engineering Education Symposium (REES)*, 2015, pp. 1-9. <https://www.researchgate.net/publication/280611224>
- [13] A-K. Carstensen, J. Bernhard, Design science research – a powerful tool for improving methods in engineering education research, *European Journal of Engineering Education*, vol. 1-18, 2018. <https://doi.org/10.1080/03043797.2018.1498459>
- [14] R. C. Richey, J. D. Klein, Developmental Research Methods: Creating Knowledge from Instructional Design and Development Practice, *Journal of Computing in Higher Education Spring*, vol. 16(2), 2005, pp. 23-38.
- [15] R. C. Richey, J. D. Klein, Design and Development Research, *Handbook of Research on Educational Communications and Technology*, vol. (141-150), Springer Science+Business Media New York, 2014. DOI: 10.1007/978-1-4614-3185-5_12
- [16] Institute of Education Sciences (IES), Common Guidelines for Education Research and Development, A Report from the Institute of Education Sciences, U.S. Department of Education and the National Science Foundation, 2013, Retrieve 17th July 2021 from <https://ies.ed.gov/pdf/commonguidelines.pdf>
- [17] M. B. Monagan, K. O. Geddes, K. M. Heal, G. Labahn, S. M. Vorkoetter, J. McCarron, P. DeMarco, *Maple 7 Programming Guide*, Waterloo Maple Inc, Canada, 2001.
- [18] Maplesoft, *User Manual*, Waterloo Maple Inc, Canada, 2021.
- [19] Kusno, I. H. Agustin, I. Halikin, On the modeling of the object surface reliefs of marble handicrafts using quartic curves and circles, *IOP Conf. Series: Earth and Environmental Science*, vol.243, 2019. DOI: <https://doi.org/10.1088/1755-1315/243/1/012017>
- [20] W.T. Li, M.C. Ho, C. Yang, A Design Thinking-Based Study of the Prospect of the Sustainable Development of Traditional Handicrafts, *Sustainability*, vol. 11(18), 2019, pp. 1-26. DOI: <https://doi.org/10.3390/su11184823>
- [21] J. Fengfan, H. Yue, Traditional Tie-dye Handicraft and Modern Design Concept Combined, *Journal of Arts & Humanities*, vol. 06(8), 2017, pp. 12-15. DOI: <http://dx.doi.org/10.18533/journal.v6i9.1230>

Pattern Recognition of Batik Madura Using Backpropagation Algorithm

Abduh Riski* Ega Bandawa Winata, Ahmad Kamsyakawuni

Mathematics Department, Faculty of Mathematics and Sciences, University of Jember, Indonesia

*Corresponding author. Email: riski.fmipa@unej.ac.id

ABSTRACT

Since October 2, 2009, UNESCO has acknowledged batik as one of Indonesia's intellectual properties. Throughout the archipelago, distinct and diverse batik motifs have emerged and been produced with the passage of time; Madura batik is one of them. The Backpropagation Algorithm is used to recognize Madura Batik Patterns in this research. Bunga Satompok, Manuk Poter, Pecah Beling, Rumput Laut, and Sekar Jagat are the motifs used in this study. To begin, resize the image to 200×200 pixels and convert it to a grayscale image. The Gray Level Co-occurrence Matrix (GLCM) approach is used to extract image features, and the Backpropagation Algorithm is used to recognize them. With GLCM, the angle orientations utilized in the feature extraction process are 0, 45, 90, and 135 degrees. There are 1, 3, and 5 hidden layers used throughout the training process, with hidden neurons in each layer of 8, 16, and 32. The highest accuracy is achieved when five hidden layers with 32 hidden neurons and one hidden layer with 32 hidden neurons in each layer are used in the testing process, which is 98 percent.

Keywords: Batik, Backpropagation, Gray level co-occurrence matrix, Neural network.

1. INTRODUCTION

Indonesia is a multicultural country that has a diversity of cultures and arts. UNESCO has designated batik as one of the world's cultural heritage from Indonesia [1]. Batik has many motifs. Almost every region in Indonesia has its motives. One of them is a motif from Madura batik. Madura batik has a characteristic colour that tends to be firm and bold. Many previous studies have done pattern recognition, for example, facial recognition [2], fingerprints [3], Retina [4], and others. The algorithm used is also diverse, one of which is the Backpropagation algorithm. Backpropagation is one of the supervised algorithms on Artificial Neural Networks that can train networks to learn patterns used during training and recognize patterns of input similar to those used during training during testing [5]. In the recognition process, the Gray Level Co-occurrence Matrix (GLCM) method is used as a feature extraction method using four GLCM computation features, namely Contrast, Energy, Correlation, and Homogeneity. The results obtained in the feature extraction process will be used to calculate the accuracy generated as a measure of the algorithm, and the method used runs well.

2. BACKPROPAGATION ALGORITHM

Artificial neural network is a computational method with an interconnected network, so it is similar to human neural networks. Artificial neural networks have an adaptive nature, which can learn from previous data and recognize data that changes [4]. One of the most widely used artificial neural network algorithms is backpropagation.

Backpropagation is an algorithm that has a strong basic mathematical [5]. Backpropagation is an algorithm that uses weight adjustment to achieve the minimum error value between predicted output and actual output [6].

In Figure 1, it can be seen that what acts as the input layer is X, and Z as the hidden layer with V is the weight between the input layer and the hidden layer. Y is the output layer, W is the weight between the hidden and the output layers, and 1 is the bias.

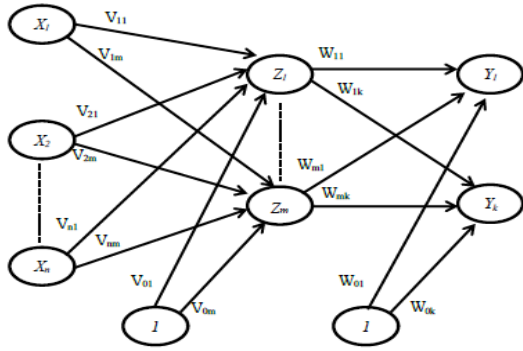


Figure 1 Backpropagation architecture.

2.1. Backpropagation Training Algorithm

The Backpropagation Artificial Neural Network training algorithm is as follows:

Step 0: Set the initial weight and bias to the lowest possible random value, set the minimum error.

Step 1: While the stopping condition is false, do steps 2 - 9.

Step 2: For each pair of training data, do steps 3-8.

Step 3: For each input unit ($x_i, i = 1, 2, \dots, n$), receive input x_i and spread the signal throughout the hidden layer neurons.

Step 4: Each hidden unit ($z_j = 1, 2, \dots, p$) sums the weight of the input signal with the following equation.

$$z_{in_j} = V_{0j} + \sum_{i=1}^p x_i v_{ij}$$

with: v_{0j} = initial bias between the input layer and the hidden layer.

v_{ij} = initial weight between the input layer and the hidden layer.

Then calculate the output value using the activation function used with the following equation.

$$z_j = f(z_{in_j})$$

And sends a signal to all output layer units.

Step 5: Each unit of output neuron ($y_k, k = 1, 2, \dots, m$) sum the weight of the output signal.

$$y_{in_k} = w_{0k} + \sum_{j=1}^m z_j w_{jk}$$

with: w_{0k} = initial bias between the hidden layer and output layer.

w_{jk} = initial weight between the hidden layer and output layer.

Calculate the output signal with the activation function in the following equation.

$$y_k = f(y_{in_k})$$

Step 6: For each output ($y_k, k = 1, 2, \dots, m$) that receives a target pattern that corresponds to the input pattern and

then calculates the error value with the following equation.

$$\delta_k = (t_k - y_k) f'(y_{in_k})$$

with: t_k = Target Output

Calculate the weight correction using the following equation.

$$\Delta w_{jk} = \alpha \delta_k z_j$$

with: α = learning rate used in training.

Calculate the correction bias with the following equation.

$$\Delta v_{jk} = \alpha \delta_k$$

And send the signal to the previous layer.

Step 7: Calculate the hidden unit factor based on the error in each hidden unit ($z_j, j = 1, 2, \dots, p$).

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk}$$

Factor δ for hidden layer .

$\delta_j = \delta_{in_j} f'(z_{in_j})$
calculate the weight correction with the following equation.

$$\Delta v_{ij} = \alpha \delta_j x_i$$

Calculate the bias correction with the following equation.

Step 8: Each neuron output ($y_k, k = 1, 2, \dots, m$) update bias and weight ($j = 0.1, \dots, p$) with the following equation.

$$w_{jk}(new) = w_{jk}(old) + \Delta w_{jk}$$

Each hidden unit ($z_j, j = 1, 2, \dots, p$) updates its bias and weight ($I = 0, 1, \dots, n$) with the following equation.

$$v_{ij}(new) = v_{ij}(old) + \Delta v_{ij}$$

Step 9: Test the stopping condition based on epoch and error with equation 1 [7].

$$MSE = \frac{1}{2} \sum_{j=1}^m (t_{kj} - y_{kj})^2 \tag{1}$$

3. GRAY LEVEL CO-OCCURENCE MATRIKS (GLCM)

GLCM is a matrix whose elements are the probability of the appearance of gray levels i and j from two different pixels at a distance (d) and angular direction ($\theta = [0^\circ, 45^\circ, 90^\circ, 135^\circ]$) [8]. A neighboring pixel with a distance d between them can be located in eight different directions, and this is shown in Figure 2.

To produce a concurrency matrix, first calculate the relationship between pixels based on the gray value of the grayscale image with the angular direction as shown in Figure 2, so that the resulting matrix concurrency with distance d and angle θ . Figure 3 shows how to calculate the concurrency matrix.

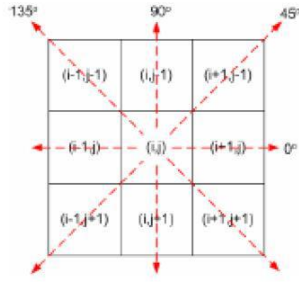


Figure 2 Neighbor relationship between pixels.

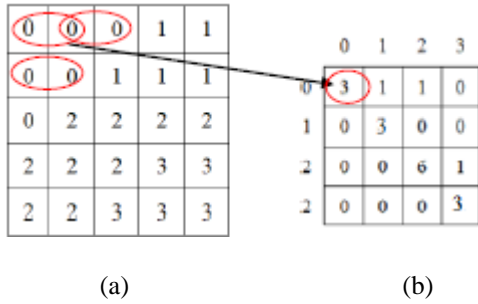


Figure 3 (a) Grayscale image values with 4 levels of gray level, (b) The concurrency matrix with the distance $d = 1$ and the angular direction $\theta = 0^\circ$.

After obtaining the concurrency matrix, the symmetric matrix is calculated by summing the concurrency matrix with the transpose, as follows.

$$\begin{bmatrix} 2 & 2 & 1 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 2 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 \\ 1 & 0 & 3 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 2 & 1 & 0 \\ 2 & 4 & 0 & 0 \\ 1 & 0 & 6 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix}$$

After obtaining the symmetric matrix, normalize the symmetric matrix by dividing each element by the sum of all elements as follows.

$$\begin{bmatrix} \frac{4}{24} & \frac{2}{24} & \frac{1}{24} & \frac{0}{24} \\ \frac{2}{24} & \frac{4}{24} & \frac{0}{24} & \frac{0}{24} \\ \frac{1}{24} & \frac{0}{24} & \frac{6}{24} & \frac{1}{24} \\ \frac{0}{24} & \frac{0}{24} & \frac{1}{24} & \frac{2}{24} \end{bmatrix}$$

After obtaining the normalization matrix, the texture value calculation is carried out using 14 GLCM features [9]. In general, research on pattern recognition using GLCM as a feature extraction method uses only five features because these five features are the main features in GLCM [10]. The five features are as follows:

- a. Entropy, the entropy value shows the randomness of the gray degree distribution of an image. Entropy is defined as:

$$\text{Entropy} = - \sum_i \sum_j P_{ij} \log P_{ij}$$

- b. Energy, commonly called ASM (Angular Second Moment), measures the uniformity of an image or value used to calculate the concentration of pair intensity on a matrix.

$$\text{ASM} = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} (P_{ij})^2$$

- c. Contrast, contrast is a measure of the spread of matrix elements in an image. Contrast is a value that is used to calculate the inaccuracy of the intensity of an image.

$$\text{Contrast} = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} (i - j)^2 \cdot P_{ij}$$

- d. Homogeneity, homogeneity or Inverse Different Moment shows the homogeneity of an image that has a similar gray.

$$\text{IDM} = \sum_{i=0}^{N_g-1} \sum_{j=0}^{N_g-1} \frac{1}{1+(i-j)^2} \cdot P_{ij}$$






- e. Correlation, correlation is a linear dependency relationship between values on the gray level in the image.

$$\text{corr} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \frac{(i-\mu_i)(j-\mu_j)p_{ij}}{\sigma_i \sigma_j}$$

4. DATA

The data used in this study are five different Madura batik motifs, where each motif will be taken as many as 30 different images. In order to obtain 150 images of Madura batik motifs. Of the 150 images divided by 100 for training images and 50 as test images. Table 1 is an example of Batik data used for each motif.

Table 1. Madura batik motifs and labeling

No.	Name of Batik	Picture	Class	Label
1	Batik Bunga Satompok		Class 1	[1;0;0]
2	Batik Manuk Poter		Class 2	[0;1;0]
3	Batik Pecah Beling		Class 3	[0;0;1]
4	Batik Sekar Jagat		Class 4	[1;1;0]
5	Batik Rumput Laut		Class 5	[1;0;1]

5. RESULT AND DISCUSSION

The recognition of Madura batik patterns with the Backpropagation Algorithm starts with the preprocessing stage, feature extraction with GLCM, and classification using the Backpropagation algorithm. The data used are five classes of Madura batik motifs taken using a smartphone camera. The preprocessing stage includes the scaling and gray scaling processes. Figure 4 is an example of an image that has been carried out in the preprocessing stage.



Figure 4 Preprocessing image.

The next step is feature extraction using the GLCM method. This stage produces values from four GLCM features with distance $d = 1$ and angular direction ($\theta = [0^\circ, 45^\circ, 90^\circ, 135^\circ]$) as in Table 2.

Table 2. GLCM value

	0°	45°	90°	135°
Contrast	0,943	1,231	0,86	1,321
Correlation	0,866	0,825	0,879	0,812
Energy	0,067	0,062	0,070	0,061
Homogeneity	0,754	0,734	0,767	0,725

The training stages use the data in Table 2 as input. Table 3 shows some of the neural network architectures that will be tested.

Table 3. Neural network architectures

Hidden Layers	Hidden Neurons in Each Layer		
	Architecture 1	Architecture 2	Architecture 3
5	32	16	8
3	32	16	8
1	32	16	8

Based on Table 3, then testing is carried out using data different from the training data. The testing results for five hidden layers and 32 neurons are displayed with the Confusion Matrix in Table 4.

Table 4. Confusion matrix

Classes	1	2	3	4	5
1	10	0	0	0	0
2	0	10	0	0	0
3	0	0	10	0	0
4	0	1	0	9	0
5	0	0	0	0	10

The accuracy is as follows:

$$\text{accuracy} = \frac{49}{50} \times 100\% = 98\%$$

The accuracy of testing from all architectures is shown in Table 5.

Table 5. Accuracy from all architectures

Hidden Layers	Hidden Neurons In Each Layer			Average
	32	16	8	
5	98%	96%	94%	96%
3	96%	96%	94%	96,33%
1	98%	96%	96%	96,67%
Average	97,33%	96%	96,67%	

The number of hidden layers and hidden neurons and the initial weights and biases used in training affect the process of image recognition. In this study, hidden layers and hidden neurons are determined manually by the researcher. In Table 5, it can be seen that the architecture using five hidden layers with 32 hidden neurons in each layer has the highest accuracy of 98%.

6. CONCLUSION

The highest accuracy is obtained in architecture using five hidden layers with 32 hidden neurons in each layer equal to 98%. The lowest accuracy is obtained in architecture using five hidden layers with eight hidden neurons and three hidden layers with eight hidden neurons equal to 94%.

REFERENCES

- [1] Y. Brasilka, Ernawati, D. Andreswari, Klasifikasi Citra Batik Besurek Berdasarkan Ekstraksi Fitur Tekstur Menggunakan Jaringan Saraf Tiruan Self Organizing MAP (SOM) (in Indonesian), Jurnal Rekursif, 2015, vol. 3, no. 2, pp. 132-145.
- [2] A. Desiani, Kajian Pengenalan Wajah dengan Menggunakan Metode Face-Arg dan Jaringan Saraf Tiruan Backpropagation (in Indonesian), Media Informatika, 2007, vol. 5, no. 2, pp. 99-111.
- [3] Sulistiani, E. Winarko, Klasifikasi Pola Sidik Jari Menggunakan Jaringan Saraf Tiruan Backpropagation (in Indonesian), Berkala MIPA, 2014, vol. 24, no. 4, pp. 215-224.
- [4] B. Poernomo, Y. Melita, Pengolahan Citra Digital Untuk Pengenalan Retina dengan Jaringan Saraf Tiruan Hopfield Diskrit (in Indonesian), Jurnal Ilmiah Teknologi dan Informasi ASIA, 2013, vol. 7, no. 1, pp. 37-47.
- [5] S. Redjeki, Perbandingan Algoritma Backpropagation dan K-Nearest Neighbor (K-NN) untuk Identifikasi Penyakit (in Indonesian), in: Proceedings of the Seminar Nasional Aplikasi Teknologi Informasi (SNATI), 2013.

- [6] A. A. Kasim, A. Harjoko, Klasifikasi Citra Batik Menggunakan Jaringan Saraf Tiruan Berdasarkan Gray Level Co-occurrence Matrices (GLCM) (in Indonesian), in: Proceedings of the Seminar Nasional Aplikasi Teknologi Informasi (SNATI), 2014.
- 7] L. Fausett,. Fundamental Of Neural Network: Architectur, Algoritm, and Aplication, New Jersey: Prentice Hall, 1993.
- [8] T. W. A. Putra, K. Adi, R. R. Isnanto, Pengenalan Wajah Dengan Koonkurensi Aras Keabuan dan Jaringan Saraf Tiruan Probabilistik (in Indonesian).
Jurnal Sistem Informasi Bisnis 02, 2013, vol. 3, no. 2.
- [9] H. Wibawanto, A. Susanto, T. S. Widodo, S. M. Tjokronegoro, Identifikasi Citra Massa Kistik Berdasarkan Fitur Gray-Level Co-occurrence Matrix (in Indonesian), in: Proceedings of the Seminar Nasional Aplikasi Teknologi Informasi 2008 (SNATI 2008), 2008.
- [10] X. Liu, K. Clarke, M. Herold, Popular Density and Image Texture: A Comparison Study America, Photogrammetric Engineering & Remote Sensing, 2006, vol. 72, no. 2, pp. 187-196.

Hybrid Cat-Particle Swarm Optimization Algorithm on Bounded Knapsack Problem with Multiple Constraints

Kiswara Agung Santoso*, Muhammad Bagus Kurniawan, Ahmad Kamsyakawuni,
Abduh Riski

Department of Mathematics, FMIPA, University of Jember

*Corresponding author. Email: kiswaras@gmail.com

ABSTRACT

Optimization problems have become interesting problems to discuss, including the knapsack problem. There are many types and variations of knapsack problems. In this paper, the authors introduce a new hybrid metaheuristic algorithm to solve the modified bounded knapsack problem with multiple constraints we call it modified bounded knapsack problem with multiple constraints (MBKP-MC). Authors combine two popular metaheuristic algorithms, Particle Swarm Optimization (PSO) and Cat Swarm Optimization (CSO). The algorithm is named Hybrid Cat-Particle Swarm Optimization (HCPSO). The results of the implementation of the algorithm are compared with PSO and CSO algorithms. Based on the experimental results, it is known that the HCPSO algorithm is suitable and can reach to good-quality solution within a reasonable computation time. In addition, the new proposed algorithm performs better than the PSO and CSO on all MBKP-MC data used.

Keywords: Hybrid cat-particle swarm optimization, Metaheuristic, Modified bounded knapsack problem.

1. INTRODUCTION

The knapsack problem is one of the optimization problems that are very interesting to be discussed. According to [1], the knapsack problem is defined as a problem that we have to determine some items to be put into storage media, where each item has weight and profit. The total weight must not exceed the capacity of the storage media used. The items must be selected such that the total profit can be maximized. Knapsack Problem (KP) has some types based on the problem, which are 0-1 knapsack, bounded knapsack, and unbounded knapsack [2]. Based on the objective, a number of constraints, and a number of storage media, the knapsack problem is divided into variations: single objective knapsack, multi-objective knapsack, multidimensional knapsack, multiple constraints knapsack, and quadratic knapsack [3]. There are some researches discuss the combination of the type and variation of the knapsack problem, such as multiple constraints 0-1 knapsack problem [4] [5], multidimensional 0-1 knapsack problem [6], multidimensional 0-1 knapsack problem with multiple constraints [7, 8], multidimensional bounded knapsack problem [9, 10], multiple constraints bounded knapsack problem [11]. In this study, we will discuss a new hybrid

metaheuristic algorithm. In Bounded Knapsack Problem (BKP), there is a maximum number of each item. In Modified Bounded Knapsack Problem (MBKP), we add a minimum number of each item that should be put into the storage media. And then, in this algorithm that we will discuss, we use three main constraints, weight, volume, and price. The total weight and volume must not exceed the capacity of the storage media, and the total price should not be greater than the capital. The optimization problem, included the knapsack problem, can be solved by some methods or algorithms. According to [12], a metaheuristic algorithm is an algorithm designed to solve optimization problems through an approach process that is inspired by nature, such as biology, physics, or ethology.

There are some metaheuristic algorithms that often used by some researchers, such as Particle Swarm Optimization (PSO) [13], Cat Swarm Optimization (CSO) [14], Genetic Algorithm (GA) [15, 16], Simulated Annealing (SA) [17, 18], Artificial Immune System (AIS) [19, 20], Ant Colony Optimization (ACO) [21, 22] and Tabu Search (TS) [23 – 25]. In this paper, we will focus on PSO and CSO algorithms. The PSO algorithm is known as the metaheuristic algorithm introduced by Eberhart and Kennedy in 1995, which is inspired by the

behavior of bird or fish swarm [13]. The PSO algorithm was first used to solve function optimization. After that, there are many other optimization problems solved using PSO algorithm, such as dynamic systems [26], pattern-matching task [21], traveling salesman problem [23], hybrid flow-shop scheduling [9], nonlinear differential equations [12], vehicle routing problem [3], vehicle routing problem with time windows [23]. The CSO algorithm was first proposed by Chu, et al. in 2006 [14] that is based on cat behaviors. There are two modes in the CSO algorithm for exploration and exploitation, that is seeking and tracing. Some researchers applied the CSO algorithm in many fields, such as function optimization [14], clustering [3], reliability-oriented task allocation [12], unconstrained optimization [5], artificial neural networks [16], 0-1 knapsack [11], workflow scheduling [18], non-unicast set covering [19]. Besides those two algorithms, many researchers have combined some algorithms as a hybrid. According to [20], the hybrid algorithm is more efficient for solving large-scale problems. A hybrid algorithm is also able to balance the exploration and exploitation, and able to balance the execution time and the quality of the final result.

In this paper, we will combine PSO and CSO algorithms as Hybrid Cat-Particle Swarm Optimization (HCPSO) algorithm. The algorithm will be tested for solving MBKP-MC. The performance of the algorithms is also compared with the results of the PSO and CSO algorithms.

2. MODIFIED BOUNDED KNAPSACK

The Bounded Knapsack Problem (BKP) [3] is defined as follows. A set of items $j = \{1, 2, \dots, n\}$ where each item has a weight w_j and a profit p_j . The problem that must be solved is how to choose items with some constrain so that the total profit can be maximized. The models which describe this problem can be written as formula below (BKP)

$$\text{Maximize } \sum_{j=1}^n p_j x_j \quad (1)$$

$$\text{with constrain } \sum_{j=1}^n w_j x_j \leq C \quad (2)$$

$$0 \leq x_j \leq b_j, x_j \text{ integer}, j = 1, 2, \dots, n \quad (3)$$

In this paper, we use Modified Bounded Knapsack Problem with Multiple Constraints (MBKP-MC). Besides the maximum number b_j of the identical copies of item j , we consider the minimum number a_j of each item type that should be selected. We also use three main constraints, which are weight (w_j), volume (v_j), and buy price/cost (c_j). The total weight and volume must not exceed the capacity of the storage media, and the total price should not be greater than the capital. Therefore, the problem is formalized as follows.

$$\text{Maximize } \sum_{j=1}^n p_j x_j \quad (4)$$

$$\sum_{j=1}^n w_j x_j \leq C \quad (5)$$

$$\sum_{j=1}^n v_j x_j \leq S \quad (6)$$

$$\sum_{j=1}^n w_j x_j \leq M \quad (7)$$

$$0 \leq x_j \leq b_j, x_j \text{ integer}, j = 1, 2, \dots, n \quad (8)$$

where C , S and M are the weight capacity, space/volume capacity and capital respectively.

3. THE PROPOSED ALGORITHM

3.1 The PSO Algorithm

The Particle Swarm Optimization (PSO) algorithm is one of the metaheuristic algorithms first introduced by Russell Eberhart and James Kennedy in 1995 [13]. The PSO algorithm is based on a swarm of birds or fish in nature. The PSO algorithm has been applied in most optimization areas, intelligent computation, and design/scheduling [21]. In the PSO algorithm, there are N particles and each particle represents a candidate of solution. The PSO algorithm generates the initial position, and velocity for every particle randomly in search space. Every iteration, each particle updates their velocity and position based on their own best position P_i and global best position P_g . According to the Invalid source specified, the particle's velocity is updated using Equation (9), and then the position is updated using Equation (10).

$$V_i(t+1) = \omega V_i(t) + c_1 r_1 (P_g(t) - X_i(t)) + c_2 r_2 (P_i(t) - X_i(t)) \quad (9)$$

where c_1 , c_2 are coefficient of acceleration in the interval $[0, 2]$; r_1 , r_2 are random variables with uniform distribution in the interval $[0, 1]$; ω is inertia weight used to balance the global and local search; and t , $t+1$ are current and next iteration respectively.

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (10)$$

The inertia weight ω is linearly decreased

$$\omega = \omega_{max} - t * \frac{\omega_{max} - \omega_{min}}{NumIter} \quad (11)$$

where ω_{max} , ω_{min} , respectively, are maximum and minimum weight, and $NumIter$ is a number of iterations.

3.2 The CSO Algorithm

Hunting ability to survive, but a house cat has a strong instinct interest in moving things. Although cats spend a lot of time sleeping, they have high alertness. They keep paying attention around to know if there is food or danger. Based on those two behaviors, the CSO algorithm is divided into two modes, seeking mode and tracing mode. The algorithm uses a mixture ratio (MR) to split the cats into seeking mode or tracing mode.

The seeking mode is used to model the situation while cats are sleeping. In seeking mode, there are four factors defined, seeking memory pool (*SMP*), seeking ranger of selected dimension (*SRD*), counts of dimension to change (*CDC*), and self-position considering (*SPC*).

SMP represents the size of the searching memory of each cat. Every cat will choose a new position in memory using Roulette Wheel. *SRD* is used to control changes in values of the selected dimension, which do not exceed the range. *CDC* is used to determine the number of dimensions to change. *SPC* is a Boolean variable. If the cat is the fittest individual, then *SPC* is true (1), otherwise, *SPC* is false (0). If *SPC* is true (1), the cat will create j copies new candidate position, where $j = SMP$, but if *SPC* is false (0), then the cat creates $j = SMP$ one copies and one candidate is the current position.

The tracing mode is used in the model cases where cats trace the target or prey. Every cat in tracing mode updated their position based on the velocity. In the CSO algorithm, the velocity of cats is updated based on the best position. The procedure of the CSO algorithm is described into eight steps as follows.

1. Generate initial position and velocity of N cats in search space.
2. Evaluate the fitness value and save the best position as $xbest$.
3. Divide the cats into seeking and tracing mode.
4. If the cat is in seeking mode, create a new candidate position using Equation (12), and then choose one to replace the current position using Roulette Wheel.

$$x'_{j,d} = x_{j,d} \pm SRD * r * x_{j,d} \quad (12)$$

where r is a random number and $x_{j,d}$, $x'_{j,d}$ are current and new values of dimension d respectively.

5. If the cat is in tracing mode, update the velocity using Equation (13), and then update the position using Equation (14)

$$v_{k,d}(t+1) = v_{k,d}(t) + c_1 r_1 (x_{best,d}(t) - x_{k,d}(t)) \quad (13)$$

$$x_{k,d}(t+1) = x_{k,d}(t) + v_{k,d}(t+1) \quad (14)$$

$d = 1, 2, \dots, D$

where c_1 is acceleration coefficient; r_1 is a random number; $v_{k,d}(t)$, $v_{k,d}(t+1)$, respectively are current and new velocity; and $x_{k,d}(t)$, $x_{k,d}(t+1)$ are current and new position respectively.

6. Merge the cats from seeking and tracing mode.
7. Evaluate the fitness value and update the best position.
8. Check the termination criterion. If the criterion is reached, then the algorithm is stopped; else, back to step 3.

3.3 The HCPSO Algorithm

The Hybrid Cat-Particle Swarm Optimization (HCPSO) algorithm is the new hybrid algorithm we propose in this paper. We combine the CSO and PSO that are known as good metaheuristic algorithms. In the

HCPSO algorithm, we use the full CSO scheme process with some additions and modifications. The algorithm saves the global best position and local best position like the PSO algorithm. And then, in the tracing mode, we apply the movement formula of PSO. Besides that, in the seeking mode, we use a modification of the value of the selected dimension based on the best position, and then we choose the best new candidate to replace the current position. This hybridization aims to get a better algorithm with a better convergence, and the execution time does not increase. All steps of the HCPSO algorithm to solve MBKP-MC are described as follows.

1. Generate initial position (X) of N individuals in search space $([0, 1])$ and generate velocity (V).

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1D} \\ x_{21} & x_{22} & \dots & x_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \dots & x_{ND} \end{bmatrix}, x_{kd} \in [0,1] \quad (15)$$

$$V = \begin{bmatrix} v_{11} & v_{12} & \dots & v_{1D} \\ v_{21} & v_{22} & \dots & v_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ v_{N1} & v_{N2} & \dots & v_{ND} \end{bmatrix} \quad (16)$$

where D is the number of items types.

2. Convert the position (X) into MBKP-MC solution term (Y) using Equation (18).

$$Y = \begin{bmatrix} y_{11} & y_{12} & \dots & y_{1D} \\ y_{21} & y_{22} & \dots & y_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ y_{N1} & y_{N2} & \dots & y_{ND} \end{bmatrix} \quad (17)$$

$$y_{kd} = \text{round}(x_{kd} * (b_d - a_d)) \quad (18)$$

3. Check all the constraints. Make sure that all the solutions are an infeasible area which means all the solutions must meet the MBKP-MC constraints.
4. Evaluate the fitness value (total profit) of all solutions.
5. Save the best position of each individual as C_k and the best global solution as C_g .
6. Divide the individuals into seeking and tracing modes.
7. If individuals are in seeking mode. Create copies based on their own the best position C_k Equation (19) and modify the selected dimension based on the best global solution C_g Equation (20).

$$x'_{j,d} = C_{k,d}, \quad d = 1, 2, \dots, D \quad (19)$$

$$x'_{j,d} = C_{g,d} \pm SRD * r * C_{g,d} \quad (20)$$

8. If individuals are in tracing mode. Update the velocity and position based on PSO movement as formulated in Equation (21) - Equation (22)

$$V_i(t+1) = \omega V_i(t) + c_1 r_1 (C_g(t) - X_i(t)) + c_2 r_2 (C_i(t) - X_i(t)) \quad (21)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (22)$$

9. Merge the cats from seeking and tracing mode and ensure all the positions do not exceed search space $[0, 1]$. If the solution exceeds the search space, then the solution must be transformed using Equation (23).

$$x_k = \begin{cases} \frac{x_k - \min(x_k)}{\max(x_k) - \min(x_k)}, & \text{if } \min(x_k) < 0 \\ \frac{x_k}{\max(x_k)}, & \text{if } \max(x_k) > 1 \end{cases} \quad (23)$$

10. Convert the new position (X) into the MBKP-MC solution term (Y). Check all the constraints and then evaluate the fitness value.
11. Update the best individual position C_k and the best global position C_g .
12. Check the termination criterion. If the criterion is reached, then the algorithm is stopped and the final solution is C_g in MBKP-MC solution term. But, if the criterion is not reached, go back to step 6.

4. COMPUTATIONAL RESULTS

Experimental Design

In order to study the performance of the proposed algorithm of this paper, we generate 20 MBKP-MC Instances. The test instances involve different size of problem. The numbers of items types (n) that we use are 30, 40, 50, 75 and 100. We apply simplex method to know the optimal value of the generated Instances. In particular, we use the following parameter setting: $N = 100$, $NumIter = 5000$, $MR = 0.5$, $SMP = 5$, $CDC = 0.2$ (30, 40, 50 items), $CDC = 0.1$ (75 items), $CDC = 0.08$ (100 items), $SRD = 0.1$, $c1 = 0.5$, $c2 = 1.5$, $\omega_{max} = 0.9$ and $\omega_{min} = 0.7$.

Table 1. The average results obtained

2.Problem Set	Opt	HCPSO		PSO		CSO	
		Z	Time	Z	Time	Z	Time
Mbkgp-mc_30_1	1095500	1095500	49.861	1095500	16.7622	1082000	57.9483
Mbkgp-mc_30_2	1174900	1174200	50.4247	1173500	16.91	1159500	56.1688
Mbkgp-mc_30_3	997100	996300	51.0127	995700	16.8141	987900	58.3414
Mbkgp-mc_30_4	1300800	1300800	47.6789	1300800	16.3793	1273100	56.1159

The average results on Table 1, that the HCPSO algorithm is the best algorithm, followed by PSO and then CSO. The PSO algorithm has the shortest execution time. Although, the average execution time of the HCPSO algorithm is less than CSO on all problem sets. These results mean that the HCPSO algorithm is suitable and can reach to good-quality solution within a reasonable computation time. The formula (24) is used to calculate the percentage of deviation (PD) and evaluate the improvement rates of the approach. The smaller PD indicates a better result because it approaches the optimal value.

$$PD = \frac{Opt - Z}{Opt} \times 100\% \quad (24)$$

where Opt is the optimal value and Z denotes the result of the algorithm.

The percentage of deviation of the algorithms for each problem is shown in Table 2.

Table 2. The percentage of deviation

Problem Set	PD		
	HCPSO	PSO	CSO
Mbkgp-mc_30_1	0.058421	0.181652	1.853035
Mbkgp-mc_30_2	0.175334	0.433228	2.279343
Mbkgp-mc_30_3	0.143416	0.405175	2.321733
Mbkgp-mc_30_4	0.063807	0.164514	2.805197
Average	0.371218	0.984129	3.359516

Based on the percentage of deviation as we can see in Table 2, the HCPSO algorithm has also the smallest average compared with PSO and CSO. We can also know that the larger problem set doesn't affect the percentage of deviation HCPSO algorithm. It is different with PSO and CSO; the percentage of deviation of both algorithms increases on larger problem sets. Thus, we can use the Hybrid Cat-Particle Swarm Optimization to solve large-sized modified bounded knapsack problems with multiple constraints.

5. CONCLUSIONS

In this paper, the Hybrid Cat-Particle Swarm Optimization (HCPSO) as a combination of Particle Swarm Optimization (PSO) and Cat Swarm Optimization (CSO) algorithms, has been proposed to solve modified bounded knapsack problem with multiple constraints (MBKP-MC). This hybridization aims to accelerate the convergence to the optimal value. The proposed algorithm has been applied to the MBKP-MC data and the results have been compared with the PSO and CSO algorithms. Based on the experimental results, it is known that the HCPSO algorithm provides better performance in the two mentioned algorithms on all problem sets. Thus, it makes the HCPSO algorithm convenient to use.

REFERENCES

- [1] A. Gherboudj, A. Layeb and S. Chikhi, Solving 0-1 Knapsack Problem by A Discrete Binary Version of Cuckoo Search Algorithm, International Journal of Bio-Inspired Computation, 2012, vol. 4, no. 4, pp. 229-236.
- [2] D. Pisinger, A Minimal Algorithm for the Multiple-Knapsack Problem, European Journal of Operational Research, 1995, vol. 83, no. 2, pp. 394-410.
- [3] H. Kellerer, U. Pferschy and D. Pisinger, Knapsack Problem, Berlin: Springer, 2003.

- [4] D. K. Agrafiotis and W. Cedeno, Feature Selection for Structure-Activity Correlation Using Binary Particle Swarms, *Journal of Medicinal Chemistry*, 2002, vol. 45, no. 5, pp. 1098-1107.
- [5] M. Clerc, Discrete Particle Swarm Optimization, Illustrated by the Traveling Salesman Problem, in *New Optimization Technique in Engineering*, Berlin, Heidelberg, Springer, 2004, pp. 2019-239.
- [6] C. T. Tseng and C. J. Liao, A Particle Swarm Optimization Algorithm for Hybrid Flow-Shop Scheduling with Multiprocessor Tasks, *International Journal of Production Research*, 2008, vol. 46, no. 17, pp. 4655-4670.
- [7] M. Babaei, A General Approach to Approximate Solutions of Nonlinear Differential Equations Using Particle Swarm Optimization, *Applied Soft Computing*, 2013, vol. 13, no. 7, pp. 3354-3365.
- [8] C. Pornsing, A Particle Swarm Optimization for Vehicle Routing Problem, Rhode Island University of Rhode Island, 2014.
- [9] Z. M. Fatimah, Penerapan Algoritma Particle Swarm Optimization untuk Vehicle Routing Problem with Time Windows pada Kasus Pendistribusian Barang, Universitas Jember, Jember, 2016.
- [10] B. Santosa and M. K. Ningrum, Cat Swarm Optimization for Clustering, *International Conference of Soft Computing and Pattern Recognition*, 2009, pp. 54-59.
- [11] R. Shojaee, H. R. Faragardi, S. Alaei, and N. Yazdani, "A New Cat Swarm Optimization based Algorithm for Reliability-Oriented Task Allocation in Distributed Systems," 6th International Symposium on Telecommunications (IST), 2012, pp. 861-866.
- [12] I. Boussaid, J. Lepagnot, and P. Siarry, A Survey on Optimization Metaheuristics, *Information Sciences*, 2013, vol. 237, pp. 82-117.
- [13] R. Eberhart and J. Kennedy, Particle Swarm Optimization, *Proceeding of the IEEE International Conference on Neural Networks*, 1995, vol. 4, pp. 1942-1948.
- [14] S. C. Chu, P. W. Tsai, and J. S. Pan, Cat Swarm Optimization, *Pacific Rim International Conference on Artificial Intelligence*, 2006, pp. 854-858.
- [15] P. C. Chu and J. E. Beasley, A Genetic Algorithm for the Multi Dimension Knapsack Problem, *Journal of Heuristics*, 1998, vol. 4, no. 1, pp. 63-86.
- [16] L. V. Snyder and M. S. Daskin, A Random-Key Genetic Algorithm for the Generalized Traveling Salesman Problem, *European Journal of Operational Research*, 2006, vol. 174, pp. 38-53.
- [17] D. R. Din, Heuristic and Simulated Annealing Algorithms for Wireless ATM Backbone Network Design Problem, *Journal of Information Science & Engineering*, 2008, vol. 24, no. 2.
- [18] M. Andresen, H. Brasel, M. Morig, J. Tusch, F. Werner and P. Willenius, Simulated Annealing and Genetic Algorithms for Minimizing Mean Flow Time in an Open Shop, *Mathematical and Computer Modelling*, 2008, vol. 48, no. 7-8, pp. 1279-1293.
- [19] O. Engin and A. Doyen, A New Approach to Solve Hybrid Flow Shop Scheduling Problems by Artificial Immune System, *Future Generation Computer Systems*, 2004, vol. 20, pp. 1083-1095.
- [20] C. A. C. Coello and N. C. Cortes, Solving Multiobjective Optimization Problems Using an Artificial Immune System, *Genetic Programming and Evolvable Machines*, 2005, vol. 6, pp. 163-190.
- [21] M. Kong, P. Tian, and Y. Kao, A New Ant Colony Optimization Algorithm for the Multi Dimension Knapsack Problem, *Computers & Operations Research*, 2008, vol. 35, no. 8, pp. 2672-2683.
- [22] M. Mavrovouniotis, F. M. Muller and S. Yang, Ant Colony Optimization with Local Search for Dynamic Traveling Salesman Problems, *IEEE Transactions on Cybernetics*, 2016, vol. 47.
- [23] V. Pureza and P. M. Franca, Vehicle Routing Problems via Tabu Search Metaheuristic, *Centre De Recherche Sur Les Transports Publication*, 1991.
- [24] W. C. Chiang and R. A. Russell, A Reactive Tabu Search Metaheuristic for the Vehicle Routing Problem with Time Windows, *INFORMS Journal on Computing*, 1997, vol. 9, no. 4, pp. 417-430.
- [25] F. Glover and M. Laguna, Tabu Search, in *Handbook of Combinatorial Optimization*, Boston, MA, Springer, 1998, pp. 2093-2229.
- [26] X. Hu and R. C. Eberhart, Adaptive Particle Swarm Optimization: Detection and Response to Dynamic Systems, *Proceedings of the 2002 Congress on Evolutionary Computation, CEC'02 (Cat. No. 02TH8600)*, IEEE, 2002, vol. 2.