A Particle Swarm Optimisation approach to the Generalised Fermat Point Problem: rethinking how a problem is solved

Weng Kin Ho
National Institute of Education
Nanyang Technological University, Singapore

Chu Wei Lim
AO Studies, Singapore

Abstract

This position paper claims that the way a mathematical problem is solved depends on the technology available to the problem-solver then. Drawing on the authors' mathematical experience of finding a new solution to an old problem – the Generalised Fermat Point Problem, salient observations are drawn to illustrate how a problem solver's experience can shaped by technological affordances.

1 Introduction

A recent exploratory joint-work with the second author marks the genesis of this paper. The long and short of it is that we were re-visiting an old geometry problem called the Fermat Point Problem (original version involves 3 points in the 2D-plane) and could not satisfy ourselves with the well-known Torricelli's geometrical solution – it cannot be modified to solve the generalised Fermat Point Problem (which involves n points, $n \geq 3$). Our probe for a solution to the generalised Fermat Point Problem led us to unearth various existing solutions – each emerged from a unique time period in history and supported by distinctive technology available at that time – and, finally, landed us on our own approach that makes use of Particle Swarm Optimisation, something of the present time.

The Fermat Point Problem and its generalisation are, after all, not new. Despite this lack of novelty, two things fuelled our exploration of this old problem. Firstly, we discovered that each existing solution in the literature fails to address some aspect of the problem. Secondly, we observed that what was available and accepted as a solution to the problem depended on the mathematics and technology available at that point of time in history. In this paper, we use the word 'technology' in a rather loose manner – that is, it is meant to encapsulate both the mathematical machinery and the physical technology associated to it. For example, geometry is the body of mathematical knowledge that deals with points, lines and curves in space, while the technology associated with geometry consists of straightedge, compass, and mechanical construction devices. [1] With these two points in mind, we set off to look for a new solution that would overcome the identified short falls. As we actively *create new solutions* to an old problem, our mathematical problem solving experience was continually enriched and shaped by the technological affordances available to us in this present computer age.

This paper (a) records our problem-solving experience as we look for a new solution to the Generalised Fermat Point Problem, and (2) draws out salient connections between *how a problem can be solved* and *the technology that avails to the problem solver(s) at that moment of problem-solving*. From these connections, we hope to gain some insights into how creative and innovative mathematical thinking take place through currently available technology.

We organize this paper as follows. Firstly in Section , we state clearly the Fermat Point Problem and recall Torricelli's geometrical characterisation of the Fermat point of a triangle. After a self-contained exposition of the existing solutions of the Fermat Point Problem (and its generalisation) in Section 3, we proceed to critique the pros and cons of these solutions. With these shortcomings in mind, we present in Section 4 our solution of the Generalised Fermat Point Problem using Particle Swarm Optimisation (as the present-day technology). In Section 5, we give a short exposition on the connection between the way a problem is solved and the available technology to the problem-solver then. Based on this connection, we discuss some implications on mathematics education, e.g. mathematics problem-solving and mathematics curriculum in schools.

2 The Fermat Point Problem (FPP)

In the early 17th century, Pierre de Fermat (1601–1665) posed a problem to Evangelista Torricelli (1608-1647), asking him for the location of the Fermat point X of a given $\triangle A_1A_2A_3$, i.e., the point X in the same plane as $\triangle A_1A_2A_3$ for which the sum of its distances from each of the vertices (A_1, A_2 and A_3) is the minimum (see Figure 1). We term this the *Fermat Point Problem*, or *FPP* for short.

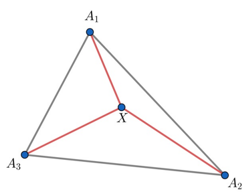

Figure 1: Determine X for which $A_1X + A_2X + A_3X$ is minimum for $\triangle A_1A_2A_3$.

The existence of the Fermat Point of an arbitrarily given triangle was first established geometrically by Evangelica Torricelli. We state and prove this result.

Theorem 1 (Fermat-Torricelli Point) *Let $\triangle A_1A_2A_3$ be given.*

[1]We therefore caution all ATMC participants to be wary of this broader sense of the word 'technology' we are using here in this paper as opposed to the usual understanding that technology refers to computer technology.

1. *If $\angle A_i \leq 120°$ for $i = 1, 2, 3$, then the Fermat Point X satisfies the condition $\angle A_1XA_2 = \angle A_2XA_3 = \angle A_3XA_1 = 120°$.*

2. *If $\angle A_i > 120°$ for some $i = 1, 2, 3$, then X is located at that obtuse angle.*

Proof. 1. First we establish that the Fermat point X, if exists, must lie inside $\triangle A_1A_2A_3$. Suppose on the contrary that X lies outside $\triangle A_1A_2A_3$. Without loss of generality, assume that X and A_1 lies on opposite sides of the line segment A_2A_3 (see Figure 2). Since X' is the

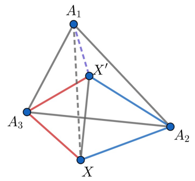

Figure 2: X' is the reflection of X in A_2A_3.

reflection of X in A_2A_3, we have that $A_2X = A_2X'$ and $A_3X = A_3X'$. Because A and X' lies on the same side of A_2A_3 and X is on the opposite side, $A_1X' < A_1X$. Consequently,

$$A_1X' + A_2X' + A_3X' < A_1X + A_2X + A_3X,$$

contradicting that X is the Fermat point of $\triangle A_1A_2A_3$. Thus, the Fermat point X, if exists, must lie inside $\triangle A_1A_2A_3$.

Suppose that $\angle A_1$ is acute. Denote by Y' the image of any given point Y under clockwise rotation of $60°$ about A_1. Then $A_1A_3 = A_1A_3'$ and $\angle A_3A_1A_3' = 60°$. Thus, $\triangle A_1A_3A_3'$ is equilateral. Now, consider a variable point X inside $\triangle A_1A_2A_3$. Then $A_1X = A_1X'$ and $\angle XAX' = 60°$. Hence $A_1X = XX'$ (see Figure 3).

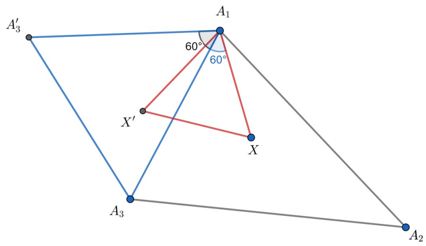

Figure 3: $\angle A_1 \leq 120°$.

Thus, we have $A_1X + A_2X + A_3X = A_2X + XX' + A_3'X'$ because $A_3X = A_3'X'$. One may further assume that $\angle A_2 \leq \angle A_3$. Note that $\angle A_2A_3A_3' \leq 180°$. Since A_2 and A_3' are given points that are fixed, the straight line distance between A_2 and A_3' is the shortest one and thus $A_1X + A_2X + A_3X = A_2X + XX' + A_3'X' \geq A_2A_3'$. In particular, equality holds if and only if X and X' lie on A_2A_3'. In other words, the minimum value of the sum $A_1X + A_2X + A_3X$ is achieved exactly when $\angle A_1XA_3' = 60°$ since in that case X' lies on $A_3'X$ (see Figure 4).

Since $\angle A_1XA_3' = \angle A_1A_3A_3' = 60°$, by the converse of the inscribed angle theorem we have that $A_1XA_3A_3'$ is a cyclic quadrilateral. Thus, X lies on A_2A_3'. In summary, X is a Fermat point of $\triangle A_1A_2A_3$ if and only if X is the intersection of the circumcircle $A_1A_3A_3'$ and

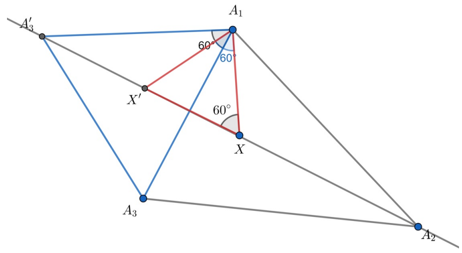

Figure 4: $A_1X + A_2X + A_3X$ is minimum exactly when X' lies on $A_3'X$.

A_2A_3'. As $\angle A_1XA_3' = \angle A_3XA_3' = 60°$, it holds that $\angle A_1XA_3 = 60°$. Similarly, $\angle A_3XA_2 = 180° - \angle A_3XA_3' = 120°$. Thus X is the required Fermat point exactly when $\angle A_1XA_2 = \angle A_2XA_3 = \angle A_3XA_1 = 120°$.

2. Without loss of generality, assume that A_1 exceeds $120°$. Again, denote by Y' the image of any given point Y under clockwise rotation of $60°$ about A_1. As in the preceding part, form $\triangle A_1A_3'A_3$ and for any point X inside $\triangle A_1A_2A_3$ form $\triangle A_1X'X$ (see Figure 5).

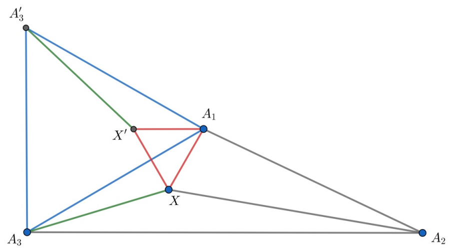

Figure 5: $\angle A_1 > 120°$.

Note that $\triangle A_1AA_3$ is congruent to $\triangle A_1X'A'3$. As before, we have that $A_1X + A_2X + A_3X = A_3'X' + X'X + XA_2$. Without loss of generality, assume that $\angle A_2 \leq \angle A_3$ so that $A_1A_3 \leq A_1A_2$. Since X is inside $\triangle A_1A_2A_3$, it holds that $X'X + XA_2 = A_1X + XA_2 \geq A_1A_2$ and so $A_3'X' + X'X + XA_2 \geq A_3'X' + A_1A_2 = A_1A_3 + A_1A_2$. Now if X lies outside $\triangle A_1A_2A_3$. Employ the reflection trick used earlier (see Figure 2), there exists a point X' insides $\triangle A_1A_2A_3$ such that $A_1X + A_2X + A_3X > A_1X' + A_2X' + A_3X'$. In our preceding consideration, since X' is inside $\triangle A_1A_2A_3$ it follows that

$$A_1X' + A_2X' + A_3X' \geq \underbrace{A_1A_1}_{=0} + A_1A_2 + A_1A_3.$$

Thus, we conclude that A_1 is the Fermat point of $\triangle A_1A_2A_3$. ∎

3 How has the FPP been solved?

Having settled the existence of the Fermat Point, We now turn our attention to the various approaches of locating it.

3.1 Geometrical technology

What kind of solutions were availed to mathematicians of Fermat and Torricelli's times? *Geometry*! Geometry, along with arithmetic, stands as one of the oldest branch of mathematics. Until the 19th century, geometry is solely devoted to Euclidean geometry which dealt with notions of points, lines, planes, angles, surfaces, and curves. Euclidean geometry has a peculiar aspect in that it can be expressly implemented by classical geometry construction relying on just a compass and straightedge (i.e., an unmarked ruler). In a way, the mathematics of geometry is defined and shaped by both the then-accepted body of geometrical axioms and results together with the then-accepted technology of the geometrical tools of those times. Both pragmatic and aesthetic in nature, geometry is a powerful tool in solving practical problems in elegance.

To locate the position of the unique Fermat Point X for the non-trivial case, that is, none of the angles A_i's exceed 120°, Torricelli offered an elegant geometrical construction: (1) Construct an equilateral triangle on each of two arbitrarily chosen sides of the triangle. (2) Draw a line from each new vertex to the opposite vertex of the original triangle. (3) The two lines intersect at the Fermat Point X (see Figure 6).

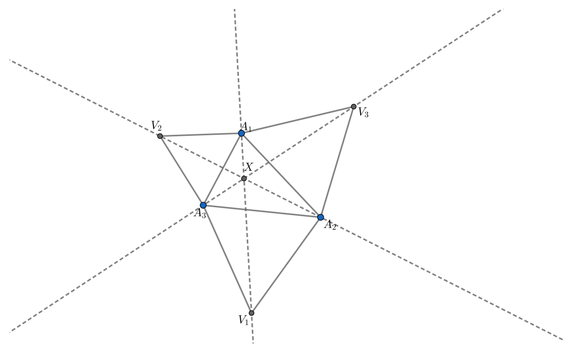

Figure 6: Constructing the Fermat Point X of $\triangle A_1A_2A_3$.

The above construction makes sense only if it results in only one point X *irrespective* of the choice of the two sides of the triangle, and this is justified below:

- $\angle V_2A_1A_2 = \angle A_3A_1V_3$; $V_2A_1 = A_3A_2$ and $A_1A_2 = A_1V_3$. $\therefore \triangle V_2A_1A_2 \equiv \triangle A_3A_1V_3$.
- $\therefore \angle A_1V_2X = \angle A_1A_3X$. By the converse of the inscribed angle theorem applied to the segment A_1X, $A_1V_2A_3X$ is a cyclic quadrilateral.
- $\angle A_1XA_3 = 180° - \angle A_1V_2A_3 = 180° - 60° = 120°$. By the inscribed angle theorem applied to the segment A_2V_3, we have $\angle A_1XV_3 = \angle A_1A_2V_3 = 60°$. So $\angle A_1XA_3 + \angle A_1XV_3 = 180°$. Thus, X lies on the line segment A_3V_3.

Therefore, X is the concurrent point of the lines A_1V_1, A_2V_2 and A_3V_3 (see Figure 7).

Remark 2 *1. Lovers of Dynamic Geometry Softwares (DSG) will be delighted to know that Geogebra, together with its automated reasoning tools, can be used to establish mathematically that the three lines A_1V_1, A_2V_2 and A_3V_3 are concurrent.*

2. There are many mathematical topics related to the Fermat Point Problem, for example, the Steiner Tree Problem. Readers who interested in the many geometrical musings related

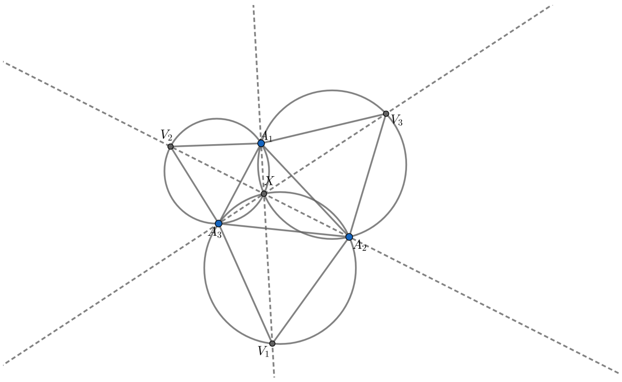

Figure 7: X is the concurrent point of lines A_1V_1, A_2V_2 and A_3V_3.

to the Fermat Point Problem and the Steiner Tree Problem may look up the recent 2010 HOGMAA Student Paper by S. Streck ([9]).

We will run through a throughout geometry (Set 1) example now. Suppose we have 3 coordinates, A_1(1,2), A_2(3,0) and A_3(0,0).

To locate the coordinates for $V_2(x_2, y_2)$, we form the following two equations:

$$\sqrt{x_2^2 + y_2^2} = \sqrt{5} \tag{1}$$

$$\sqrt{(x_2 - 1)^2 + (y_2 - 2)^2} = \sqrt{5} \tag{2}$$

Solving them simultaneously, we have $x_2 = -\sqrt{\frac{13-4\sqrt{3}}{4}} = \frac{1-2\sqrt{3}}{2}$ and $y_2 = 1 + \frac{\sqrt{3}}{2}$. Now, we proceed to find V_1. As the $x-$coordinate of V_1 is $\frac{3}{2}$, we have the following equation:

$$(3 - \frac{3}{2})^2 + y^2 = 3^2 \tag{3}$$

giving us $y = -\frac{\sqrt{27}}{2}$. Hence, V_1 is $(\frac{3}{2}, -\frac{\sqrt{27}}{2})$.

Constructing a straight line l_1 passing through $V_1(1.5, -\frac{\sqrt{27}}{2})$ and $A_1(1, 2)$ and l_2 passing through $V_2(\frac{1-2\sqrt{3}}{2}, 1 + \frac{\sqrt{3}}{2})$ and $A_2(3, 0)$, we have the following

$$l_1 : y = (-4 - 3\sqrt{3})x + 6 + 3\sqrt{3} \tag{4}$$

$$l_2 : y = -\frac{4 + \sqrt{3}}{13}x + \frac{12 + 3\sqrt{3}}{13} \tag{5}$$

Solving equations 4 and 5 simultaneously, we obtained our Fermat point $X(\frac{234+195\sqrt{3}}{507}, \frac{351+39\sqrt{3}}{507})$, in exact form.

3.2 Mechanical technology

Moving ahead into the 18th century, mechanics has developed into a fairly mature field of mathematics. In particular, the Italian-French mathematician and astronomer Joseph-Louis Lagrange introduced *Lagrangian mechanics* in his 1788 work, Mécanique analytique, which founded classical mechanics on the *stationary-action principle* (also known as the *principle of least action*). Lagrangian mechanics characterises a mechanical system as a pair (M, L) comprising a configuration space M and a smooth function L within that space called a *Lagrangian*. For most systems, $L := T - V$, where T and V the kinetic and potential energy of the system, respectively. The stationary action principle asserts that the action functional of the system derived from L must remain at a stationary point throughout the time evolution of the system. This constraint then gives rise to the equations of motion of the system via the Lagrange's equations.

The FPP can be cast into a classical mechanics problem. Construct the $\triangle A_1A_2A_3$ on a horizontal frictionless plane and place holes at each vertex A_1, A_2, A_3. Consider a system of three separate equal masses, each connected by long massless strings to a pivot point in the triangle (see Figure 8). Pass one string through one of the holes that hangs a mass positioned under each vertex, and let the system come to equilibrium.

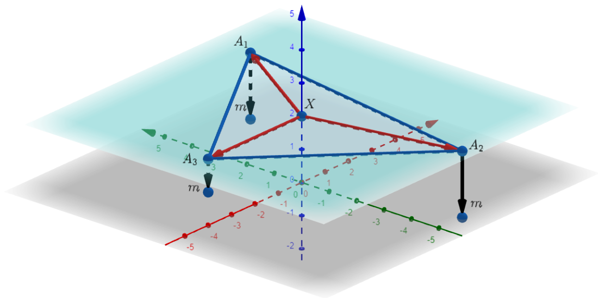

Figure 8: Mechanical set-up for FFP.

Lemma 3 *At equilibrium, the pivot point is exactly at the Fermat point X of $\triangle A_1A_2A_3$.*

Proof. By the stationary-action principle, the above mechanical system looks for the stationary state of L, where $L = T - V$, where T and V the kinetic and gravitational potential energy of the system, respectively. Since the system is at static equilibrium, $T = 0$. Hence the stationary value of L is attained when the gravitational potential energy is minimized, which can be achieved by setting the weights closest to the ground. For each weight, the length of string below the vertex would be as long as possible, or equivalently, the length of string on the triangle is as short as possible. Hence the minimum V occurs the total string lengths on the triangle is minimized – precisely when the pivoting point is at the Fermat point. ∎

Theorem 4 *The Fermat point X of $\triangle A_1A_2A_3$ satisfies the condition*

$$\angle A_1XA_2 = \angle A_2XA_3 = \angle A_3XA_1 = 120°.$$

Proof. At equilibrium, the vector sum of the three forces acting on the pivot point is zero. Since each mass is equal, these forces must be exactly 120° away from each other. ∎

3.3 Vectorial technology

A systematic study and use of vectors were a 19th and early 20th century phenomenon. With the vectorial machinery available to us, what type of solution for the FPP can emerge? A vectorial solution of the FPP was put forth by Titu Andreescu and Oleg Mushkarov ([1]). Herein, we first state a simple extension of the Fermat Point Problem to the so-called *Generalised Fermat Point Problem*, and then give a vectorial proof that extends Andreescu and Mushkarov's.

Definition 5 (Generalised Fermat Point Problem) *Let $n \geq 3$ be a positive integer. Given n distinct co-planar points $A_1, \cdots, A_n$, locate a point X (if it exists), co-planar with the points A_i's, for which*

$$A_1X + \cdots + A_nX$$

is minimum. We use the acronym GFPP *for* Generalised Fermat Point Problem.

Theorem 6 *The point X co-planar to the given points $A_1, A_2, \ldots, A_n$, where $n \geq 3$, for which the unit vectors from X to each of the A_i's sum up to the zero vector is the Fermat point of $A_1, A_2, \ldots, A_n$.*

Proof. Let X be such a point. Denote by $\mathbf{a}_i$ the position vector of A_i with respect to X, and by $\mathbf{u}_i$ its corresponding unit vector. By the definition of dot product of vectors, it holds that $|\mathbf{a}_i| = |\mathbf{a}_i||\mathbf{u}_i| \cos 0 = \mathbf{a}_i \cdot \mathbf{u}_i$ for all $i = 1, 2, \cdots, n$.

Pick an arbitrary point P coplanar with $A_1, A_2, \ldots, A_n$, and let its position vector with respect to the Fermat point X be $\mathbf{p}$. Thus, for each $i = 1, 2, \ldots, n$, we have that

$$|\mathbf{a}_i| = \mathbf{a}_i \cdot \mathbf{u}_i = (\mathbf{a}_i - \mathbf{p}) \cdot \mathbf{u}_i + \mathbf{p} \cdot \mathbf{u}_i \leq |\mathbf{a}_i - \mathbf{p}||\mathbf{u}_i| + \mathbf{p} \cdot \mathbf{u}_i = |\mathbf{a}_i - \mathbf{p}| + \mathbf{p} \cdot \mathbf{u}_i.$$

Adding all these quantities together, we arrive at

$$|\mathbf{a}_1| + \cdots + |\mathbf{a}_n| \leq |\mathbf{a}_1 - \mathbf{p}| + \cdots + |\mathbf{a}_n - \mathbf{p}| + \mathbf{p} \cdot (\mathbf{u}_1 + \cdots + \mathbf{u}_n).$$

Since X satisfies the condition $\mathbf{u}_1 + \cdots + \mathbf{u}_n = \mathbf{0}$, it follows that $|\mathbf{a}_1| + \cdots + |\mathbf{a}_n| \leq |\mathbf{a}_1 - \mathbf{p}| + \cdots + |\mathbf{a}_n - \mathbf{p}|$, which is equivalent to

$$A_1X + \cdots + A_nX \leq A_1P + \cdots + A_nP.$$

Since P is an arbitrary point coplanar with A_i's, X is the Fermat point of A_i's. ■

Corollary 7 (GFPP for $n = 4$) *Let A_1, A_2, A_3 and A_4 be four distinct coplanar points. Then the Fermat point X of these four points is exactly the intersection of the diagonals of the quadrilateral $A_1A_2A_3A_4$.*

4 How can the FPP be solved now?

In this section, we briefly review each of above existing solutions of the FPP. Arguing that all these solutions were inspired and supported by technology available at specific points of time, we show how the GFPP can be solved in yet different ways by exploiting *given* technology of *present times*. The technology we use is that of Particle Swarm Optimisation (PSO).

4.1 Discussion about the existing solutions

Torricelli's solution relied on geometrical construction which is expectedly the approach used during his time. This is particularly so since a geometrical problem naturally asks for a geometrical solution. Indeed, Fermat anticipated this from Torricelli. Examining the details presented in Theorems 1 and 4, one must agree, in spite of the ingenuity of the proof and the aesthetics of the geometrical construction, that the manner of locate the Fermat point is way too complicated, and fairly tedious using compass and straightedge construction. More importantly the Torricelli's construction cannot be easily extended to solve the n-point GFPP ($n > 3$).

The proof inspired by Lagrangian mechanics is remarkably terse and leverages on the somewhat mystical prowess of theoretical mechanics. Furthermore, the technology is visible: you can set up a physical experiment using weights, strings and a triangle lamina to locate of the Fermat Point with a bit of trial-and-error. "Is this a mathematically legitimate solution?", you may ask. We do not have an answer here, but might just want to borrow a remark enunciated by Philip Davis in his paper "When Is A Problem Solved?":

> Is such and such really a solution? ... Apparently, the meaning of the word "solution" can be stretched quite a bit. The elastic quality of mathematical terms or definitions is remarkable, and is often achieved through context enlargement. ([3])

Unlike the two preceding solutions, the vectorial technology is pegged at a desirable level of abstraction that simultaneously veers us away from the geometrical intricacy and requires only a light mathematical overhead as compared to Lagrangian mechanics. Despite its advantages, the vectorial method is not constructive, i.e., it does not compute the position vector of the Fermat point X in terms of the given points A_i's – many would agree that this is a major defect of the vectorial method in contrast to Torricelli's geometrical method and the mechanical method. Nevertheless, the zero vector sum criterion provides us a computable test of accuracy for any candidate Fermat point.

Given the present-day computer technology, would we be able to create an alternative solution to the GFPP that can possibly help us overcome the shortcomings we have discussed so far. In the next subsections, we develop the theory and implementation of our proposed modern technology.

4.2 A short primer on PSO

Particle Swarm Optimisation (PSO, for short) refers to a characteristic class of mathematical algorithms – first created by James Kennedy and Russell Eberhart ([5]) – inspired by the movements of birds flying in a flock, where members of the flock benefit from the experience of other members. For instance, when searching for food a flock of birds can share their discoveries and the entire flock can use the information to locate the place where more food can be found. The individual bird will then fly in a vector which is based on information from the flock and information it has gathered individually. This vector changes every day, as the flock finds better hunts, and the individual bird also finds better hunts. Eventually, all the flock of birds will converge on a single point, which is the position of the best hunt obtainable. Our idea is to exploit the swarm movement to solve the GFPP, where particles move as a swarm and eventually converge to the Fermat point.

Certain mathematical terminologies used in PSO must first be established. Imagine a swarm of birds (swarm particles) flying in the same plane as the polygon $A_1 \ldots A_n$, where $A_1, \ldots, A_n$ are distinct points having position vectors $\mathbf{a}_k$ $(k = 1, \ldots, n)$ in search of the Fermat point X with position vector $\mathbf{x}$, i.e., where the *fitness function* or *objective function* $f : \mathbb{R}^2 \longrightarrow \mathbb{R},\ f(\mathbf{x}) = \sum_{k=1}^{n} |\mathbf{x} - \mathbf{a}_k|$ is to be minimised. This function f calculates the sum of the distances between X and each of the vertices $A_1, A_2, \ldots, A_n$.

A swarm consists of N birds, and each ith bird archives an aerial log of its position vector $\mathbf{x}_i^d$ on the dth day. Given that the daily velocity $\mathbf{v}_i^d$ of the ith bird on the $(d+1)$th day, the daily position vector $\mathbf{x}_i^d$ of the ith bird is determined recursively by $\mathbf{x}_i^{d+1} = \mathbf{x}_i^d + \mathbf{v}_i^{d+1}$ for $i = 1, \ldots, N$. The key business in PSO is to model the daily velocity of each bird, which is determined by three components: (i) *inertia* component, (ii) *cognitive* component, and (iii) *social* component. We explain this now.

Inertia component. The ith bird's velocity, denoted $\mathbf{v}_i^d$, gives rise to a certain scalar multiple $w\mathbf{v}_i^d$ of it which is then its *inertia component* of the velocity on the $(d+1)$th day. This inertia component represents the ith bird's degree of reluctance to change its previous day's velocity.

Cognitive component. Each ith bird keeps a record of its *personal best position*, i.e., the position vector $\mathbf{P}_i^d$ at which the smallest value of the objective function f has been achieved up to and including the dth day. A randomized scaling of the directional vector $\mathbf{P}_i^d - \mathbf{x}_i^d$ models the *cognitive component* of the velocity of the ith bird on the $(d+1)$th day.

Social component. Each ith bird also keeps in its aerial log book the *global best position*, i.e., the position vector $\mathbf{G}^d$ of the location found by one of the N birds in the course of the swarm's flight up to and including the dth day. A randomized scaling of the directional vector $\mathbf{G}^d - \mathbf{x}_i^d$ models the *social component* of the velocity of the ith bird on the $(d+1)$th day.

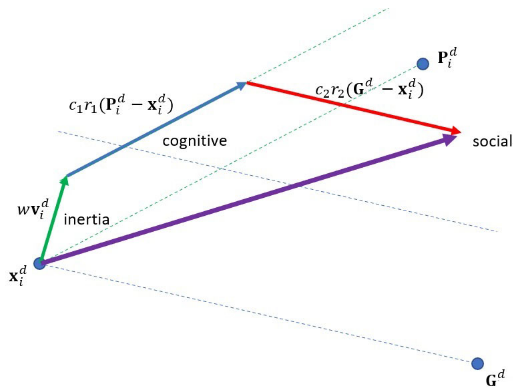

Figure 9: Daily velocity comprises three components.

Thus, the velocity $\mathbf{v}_i^{d+1}$ is the resultant of the aforementioned component vectors:

$$\mathbf{v}_i^{d+1} = \underbrace{w\mathbf{v}_i^d}_{\text{inertia}} + \underbrace{c_1 r_1(\mathbf{P}_i^d - \mathbf{x}_i^d)}_{\text{cognitive}} + \underbrace{c_2 r_2(\mathbf{G}^d - \mathbf{x}_i^d)}_{\text{social}}, \tag{6}$$

where $i = 1, \ldots, N$ (see Figure 9). Certain constants, parameters, and randomized numbers appear in the above recursive equation, whose details are described below:

1. The bounded random constant w in the term $w\mathbf{v}_i^d$ tunes the scope of exploration of the ith bird, modelling the distance it covers each day. A large value of w signifies that the bird leans towards exploring further along yesterday's velocity.
2. The parameter c_1 in the term $c_1 r_1(\mathbf{P}_i^d - \mathbf{x}_i^d)$ is the impact factor of the cognitive component. An increase in the value of c_1 indicates a higher cognitive impact, i.e., the velocity of flight is influenced more by the bird's own belief system based on its own personal best records. The parameter c_2 in the term $c_2 r_2(\mathbf{G}^d - \mathbf{x}_i^d)$ is the impact factor of the social component. An increase in the value of c_2 indicates a higher social impact, i.e., the velocity of flight is influenced more by the swarm's global bests.
3. r_1 and r_2 are just random numbers uniformly distributed in the unit interval $[0,1]$.

For more detailed account of the theory and applications of PSO, readers may refer to **??**.

4.3 Sample runs of PSO algorithm

Equation (6) is a recursive equation that can be easily implemented in PYTHON. The PYTHON codes[2] of the PSO algorithm for locating the Fermat point of the set of points $A_1,\ A_2,\ldots,A_n$ $(n \geq 3)$ can be found at the Github repository (`https://github.com/howengkin/pso.io.git`).

We ran the PSO algorithm over many sets of data – different sets of points (with various sizes). Figures 10 and 11 shows animation clips of a swarm movement observed in a particular run of this PSO algorithm for two sample sets of data:

1. Set 1.
   - Number of points: 3
   - Data set: A_1(1,2), A_2(3,0) and A_3(0,0).
   - Number of swarm particles: 100
   - Number of iterations: 100
   - PSO Fermat point: $X(1.12771183, 0.8255423)$
   - Exact Fermat point:$\frac{234+195\sqrt{3}}{507}, \frac{351+39\sqrt{3}}{507}$
   - Percentage error by Euclidean metric:

$$x - \text{coordinate} : \left| \frac{1.12771183 - \frac{234+195\sqrt{3}}{507}}{\frac{234+195\sqrt{3}}{507}} \right| \times 100\% = 0.00000169\%$$

$$y - \text{coordinate} : \left| \frac{0.8255423 - \frac{351+39\sqrt{3}}{507}}{\frac{351+39\sqrt{3}}{507}} \right| \times 100\% = 0.00000846\%$$

   - Sum of unit vectors $\mathbf{u}_i$:

$$\mathbf{u}_{A_1X} + \mathbf{u}_{A_2X} + \mathbf{u}_{A_2X}$$

$$= \begin{pmatrix} 0.1081038411 \\ -0.9941396077 \end{pmatrix} + \begin{pmatrix} -0.9150020989 \\ 0.4034490787 \end{pmatrix} + \begin{pmatrix} 0.8068982404 \\ 0.5906904685 \end{pmatrix} = \begin{pmatrix} -0.0000000174 \\ -0.0000000605 \end{pmatrix}$$

[2]Readers may also make an email request for the PYTHON codes from the authors.

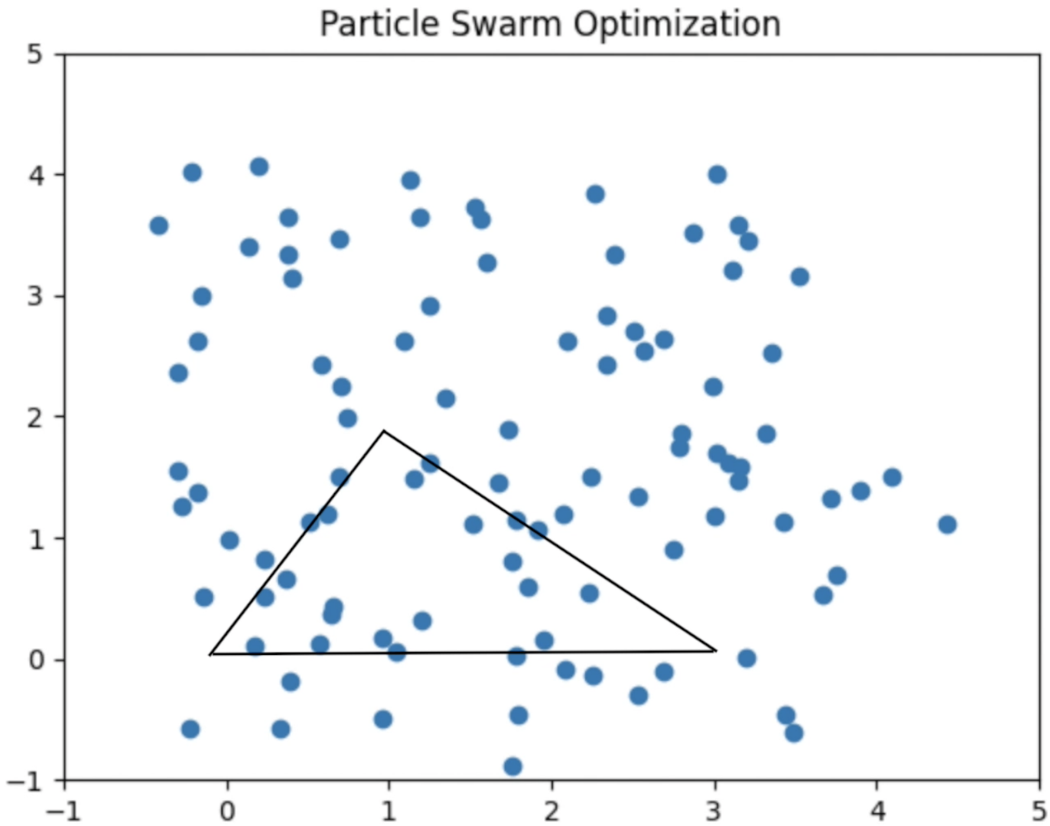

Figure 10: Swarm movement for Set 1.

2. Set 2.

- Number of points: 5
- Data set: $A(2,1), B(1,2), C(2,2.5), D(3,2), E(3,1)$.
- Number of swarm particles: 100
- Number of iterations:100
- PSO Fermat point: $Y(2.26885941, 1.67413429)$
- Exact Fermat point:
- Percentage error by Euclidean metric:
- Sum of unit vectors $\mathbf{u}_i$:

$$\mathbf{u}_{AY} + \mathbf{u}_{BY} + \mathbf{u}_{CY} + \mathbf{u}_{DY} + \mathbf{u}_{EY}$$

$$= \begin{pmatrix} 0.3704470089 \\ 0.9288536018 \end{pmatrix} + \begin{pmatrix} 0.9685688202 \\ -0.2487457348 \end{pmatrix} + \begin{pmatrix} 0.3095578354 \\ -0.9508806164 \end{pmatrix} + \begin{pmatrix} -0.9133871240 \\ -0.4070920802 \end{pmatrix} + \begin{pmatrix} -0.7351865730 \\ 0.6778648117 \end{pmatrix} = \begin{pmatrix} -0.0000000325 \\ -0.0000000178 \end{pmatrix}$$

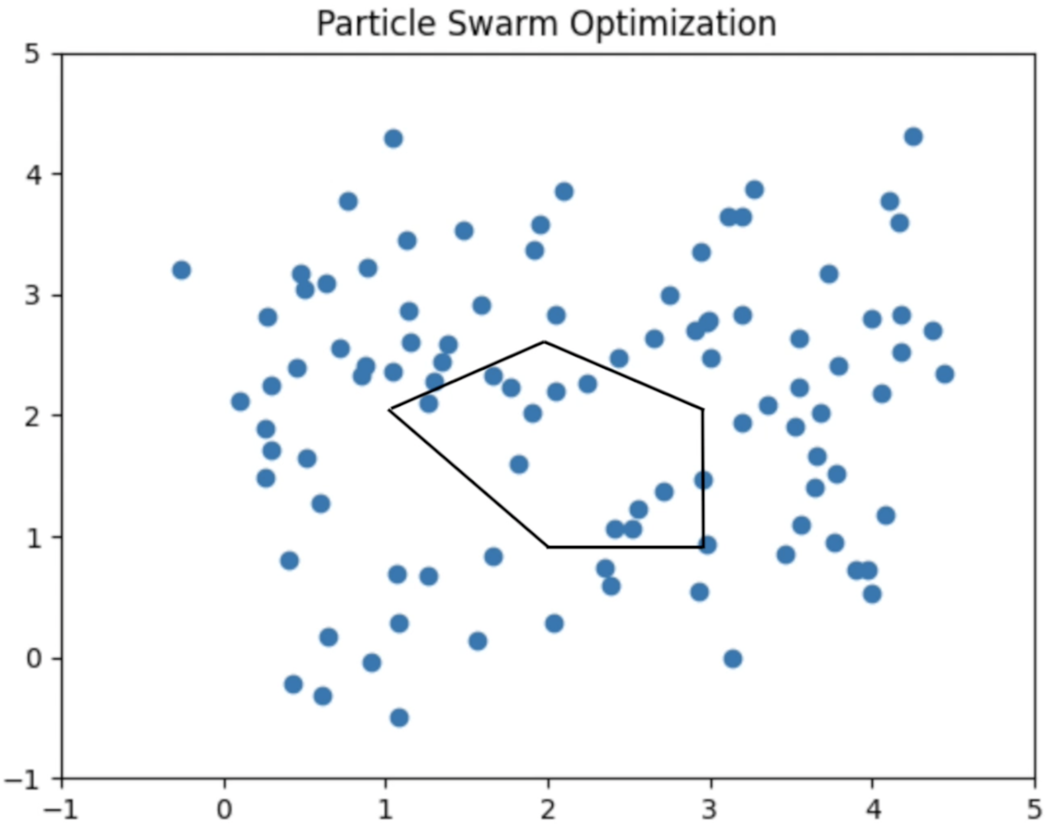

Figure 11: Swarm movement for Set 2.

4.4 Performance evaluation

We summarise the performance evaluation of the PSO algorithm for GFPP here.

Few design parameters. The PSO algorithm involves relatively few parameters, and this makes the programming task very simple. This has the advantage that anyone who wants to learn and apply the PSO method finds the codes easy to understand or write.

Concurrent applications of efficient global search. Running the PSO algorithm concurrently over many data sets of various sizes, we observed that the global search for the Fermat point in each of these runs is very efficient and accurate with convergence occurring under 100 iterations on the average. While the PSO method cannot be strictly classified as constructive because the point of convergence of the swarm is not the exact Fermat point in general, it nonetheless is very fast in locating a highly accurate approximation to the actual Fermat point, and would have sufficed in most practical situations.

Problem Solving Component Resource	Impact of (Computer) Technology Dramatic increase in the accessible knowledge base
Heuristics	More opportunities for effective use of heuristic: making a table, drawing a graph
Control	Provides more effective management strategies
Beliefs	Develops beliefs that are specific to the context in which (computer) technology was used

Table 1: Contribution of the (computer) technology to problem solving.

Insensitive to scaling of design parameters. In cases where the given set of points $A_1, A_2, \ldots, A_n$ are spaced far apart, appropriate scaling incorporated into the PYTHON codes did not dampen the speed of convergence to the Fermat point(s). This suggests that the PSO method is insensitive to scaling of the variables.

Derivative-free and naturalistic. The PSO solution is completely different in nature from the geometrical solution of Torricelli. Compared to the mechanical method, there is one commonality. That is, both rely on some physical phenomena to get as close to the exact Fermat point as possible: the PSO method exploits the collective wisdom of the swarm particles to search for the solution while the mechanical method relies on the forces in the system and some careful adjustments of the pivoting point. Another important characteristic of the PSO method is that it is derivative-free, i.e., it does not make use of any differentiation techniques in the whole process.

5 How can a problem be solved?

According to Alan Schoenfeld ([7]), four categories of knowledge and skills are required to be successful in mathematics problem solving: (1) resources – proposition and procedural knowledge of mathematics, (2) heuristics – strategies and techniques for problem solving such as working backwards, drawing figures, (3) control – decisions about when and what resources and strategies to use, and (4) beliefs – a mathematical "world view" that determines how the problem-solver approaches a problem. To these four categories, we propose to add on one more: *technology.*

Already, in [4], the impact of technology (in particular, computer technology) on each of these above categories has long been identified (see Table 5, reproduced/modified from [4, p.4]).

One important reality teased out from [4, p.5] is that "problem-solving strategies are dependent on the context of the problem, goal and motives of the problem-solver, and the *accessible tools*[3]" This reality is of particular relevance to us in this paper concerning our experience of solving the GFPP using PSO, and we develop it further.

[3]The emphasis has been added by the authors of this paper.

Context of the problem. The GFPP involves optimisation with a geometrical backdrop. The task of GFPP focuses on locating the Fermat point, but does not restrict the problem-solver to employing only calculus techniques and/or geometrical constructions. Additionally, optimisation problems occurring in practice demand efficiency in obtaining an approximate solution that satisfies a certain level of accuracy, not necessarily an exact one.

Goals and motives of the problem-solver. In the case of GFPP, we wanted to look for an alternative solution while having a relatively light theoretical overhead must be versatile and quick enough to locate the Fermat Point, given more than 3 points. It is the problem-solver's goal to overcome the shortcomings of the existing solutions using this new solution.

Accessible tools. To be able to meet the goals and motives of the problem-solver, it is of paramount importance that a suitable tool is in existence and made available to the problem-solver. Notice that the existing method of vectors already prepared the way for the PSO method since the search space can be taken as the 2D plane, and the swarm particles are nothing but probes moving in the search space and searching for the point of optimisation. The computer that processes the PSO algorithm gives the problem-solver the required speed and accuracy. The problem-solver did not develop the PSO method but knows of it – this knowledge is counted as problem-solving resource, and applies it as a powerful tool which is suitable to satisfy the requirements of the task set in the GFPP.

Remark 8 *For more recent works of the role of technology on mathematics education, we point the reader to [2].*

6 Conclusion

Technology has improved the teaching and learning of mathematics over the past decades by actively transforming pedagogy, policy and practice. These transformations are informed by a growing body of meta-analytic research examining how the affordances of technology shape mathematical experiences ([11]). Our experience in applying the Particle Swarm Optimisation method in locating the Fermat point in the context of the Generalised Fermat Point Problem reinforces the claim that the way a problem is solved crucially depends on the accessible technology that came at the right time and the right place to meet certain goals and motives of the problem solver, given the context of the problem itself.

There remain skeptics who would question whether our way of solving the GFPP using PSO is even mathematically legitimate in the first place. After all, it looks like the PSO method is a guess-and-check method that has been performed quickly by a computer! Are students, teachers and mathematicians ready to enlarge the context of understanding what a solution to a mathematics problem so as to embrace the thesis that technology – when it becomes accessible to the problem-solver – will, in part, determine the way how a problem is solved? An answer to this question will fundamentally change the way we teach and learn mathematics at schools, allowing us to be more open-minded in applying available technology in creative and innovative ways!

References

[1] AoPS Online. *Fermat Point.* Available at https://artofproblemsolving.com/wiki/index.php/Fermat_point.

[2] Cullen, C., Hertel, J., and Nickels, M. (2020). "The Roles of Technology in Mathematics Education", *The Educational Forum*, 84(3), pp. 1-13.

[3] Davis, P. J. (2008). "When Is a Problem Solved?", Chapter in *Proof and Other Dilemmas*, Mathematics and Philosophy, pp. 81-94. Mathematical Association of America.

[4] Jurdak, M. (2000). "Technology and Problem Solving in Mathematics: Myths and Reality", In *Proceedings of the International Conference on Technology in Mathematics Education* (July 5-7, 2000, Beirut, Lebanon). pp. 30-37.

[5] Kennedy, J., Eberhart, R. C. and Shi, Y. (2001). *Swarm Intelligence*, Academic Press, United States of America.

[6] Seyedali Mirjalili S. M. Mirjalili and A. Lewis (2014). *Grey wolf optimizer*, Advances in Engineering Software 69, pp. 46-61.

[7] Schoenfeld, A. (1985). *Mathematical Problem Solving.* New York: Academic Press.

[8] Sharma, K., Chhamunya, V., Gupta, P. C., Sharma, H., and Bansal, J. C. (2015). *Fitness based Particle Swarm Optimization*, Rajasthan Technical University.

[9] Streck, S. (2020). "The Fermat Problem", *HOM SIGMAA 2010 Student Paper Contest.* Available at https://maa.org/sites/default/files/pdf/upload_library/46/HOMSIGMAA2010/StreckHOM2010.pdf.

[10] Tam, A. (2022). *A Gentle Introduction to Particle Swarm Optimisation*, Machine Learning Mastery, https://machinelearningmastery.com/a-gentle-introduction-to-particle-swarm-optimization/ (accessed October 11, 2022)

[11] Young, J. (2017). *Technology-enhanced mathematics instruction: A second-order meta analysis of 30 years of research*, Educational Research Review 22, pp. 19-33.