10 (2) July 2024 116-126
Register: *Jurnal Ilmiah Teknologi Sistem Informasi*

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)

Contents lists available at www.journal.unipdu.ac.id

Register

Journal Page is available to www.journal.unipdu.ac.id/index.php/register

Research article

Utilization of the Particle Swam Optimization Algorithm in Game Dota 2

Hendrawan Armanto [a], Harits Ar Rosyid [b,], Muladi Muladi [c], Gunawan Gunawan [d]*

[a,b,c] *Program Studi Teknik Elektro dan Informatika, Universitas Negeri Malang, Malang, Indonesia*
[a,d] *Program Studi Informatika, Institut Sains dan Teknologi Terpadu Surabaya, Surabaya, Indonesia*
email: [a] hendrawan.armanto.2205349@students.um.ac.id, [b,*] harits.ar.ft@um.ac.id, [c] muladi@um.ac.id, [d] gunawan@stts.edu
* *Correspondence*

ARTICLE INFO

Article history:
Received 2 Mei 2023
Revised 1 August 2024
Accepted 29 October 2024
Available online 11 Nopember 2024

Keywords:
MOBA
Dota 2
Artificial Intelligence
Particle Swarm Optimization

Please cite this article in IEEE style as:
H. Armanto, H. A. Rosyid, M. Muladi, and G. Gunawan, "Utilization of the Particle Swam Optimization Algorithm in Game Dota 2," *Register: Jurnal Ilmiah Teknologi Sistem Informasi*, vol. 10, no. 2, pp. 116-126, 2024.

ABSTRACT

Dota 2, a Multiplayer Online Battle Arena game, is widely popular among gamers, with many attempting to create efficient artificial intelligence that can play like a human. However, current AI technology still falls short in some areas, despite some AI models being able to play decently. To address this issue, researchers continue to explore ways to enhance AI performance in Dota 2. This study focuses on the process of developing artificial intelligence code in Dota 2 and integrating the particle swarm optimization algorithm into Dota 2 Team's Desire. Although particle swarm optimization is an old evolutionary algorithm, it is still considered effective in achieving optimal solutions. The study found that PSO significantly improved the AI Team's Desire and enabled it to win against Default AI of similar levels or players with low MMR. However, it was still unable to defeat opponents with higher AI levels. Furthermore, this study is expected to assist other researchers in developing artificial intelligence in Dota 2, as the complexity of the development process lies not only in AI but also in language, structure, and communication between files.

1. Introduction

Games are becoming increasingly significant in society, serving not only as entertainment purposes but also for education and skill development, especially among children. This trend is driven by rapid advancements in technology and game content. As a result, research in the field of gaming has also expanded, including areas such as game theory [1]–[3], game design [4]–[6], and artificial intelligence [7]–[10]. However, each study in this field is often dependent on the specific genre or type of game being studied, as each game possesses unique concepts, mechanics, and settings. Among the most popular game genres today is the Multiplayer Online Battle Arena (MOBA) [11]. First introduced in 2003, MOBA games gained immense popularity in 2016, especially on mobile platforms, which is due to improved mobile technology. Currently, there are around 40[1] MOBA games, each with its distinctive features[2].

Although MOBA games are a relatively new genre, they are actually a subset of the long-established real-time strategy (RTS) genre [12]. In RTS games, players focus on using characters' or teams' abilities to develop effective strategies for victory. Different games offer various strategies and approaches, such as resource acquisition, unit formation, and unit positioning. As a branch of RTS, MOBA games also demand well-planned strategies to succeed. Common tactics in MOBA games include character selection, player movement, attack coordination, item choice, and skill selection. This strategic complexity makes MOBA games both exciting and challenging to play.

[1] Based on Wikipedia on the link https://en.wikipedia.org/wiki/List_of_multiplayer_online_battle_arena_games. However, because Wikipedia is a knowledge website that is inputted by various groups of people and not experts, the accuracy does not reach 100%.
[2] Although every game has its unique features, the MOBA genre must maintain the concept of fairness [34]. Without fairness, the game will be unbalanced, resulting in one-sided gameplay.

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
10 (2) July 2024 116-126

Dota 2 is one of the most popular MOBA games, boasting a massive fanbase, including international communities and E-Sports tournaments [13]. However, not all players can compete at a high level, so many opt for the game mode against artificial intelligence or "BOTs" to practice individually or with their teams. Although various communities have developed AI for Dota 2, only a few effectively compete against experienced players. To date, only one officially published study [14], [15] has successfully created an AI capable of playing at a professional level, but it made the game too challenging for general enjoyment. Developing AI in Dota 2 poses significant challenges due to the less commonly used programming language, complex file structure, and intricate game strategies. Consequently, research on AI in Dota 2 is limited, with most studies focusing on predicting victory based on hero selection [16] or analyzing gameplay patterns [17].

Particle Swarm Optimization (PSO) is used in this study for two main reasons: its balance between exploitation and exploration[3] and its support for integer representations, which are essential here. Despite being a well-established algorithm, PSO remains widely used across various research fields [18]–[23]. Compared to the newer evolutionary algorithms [24], PSO is notably popular, partly because it is relatively easy to learn and apply to different problems. Using PSO as the primary algorithm for developing artificial intelligence in Dota 2 is a logical choice, given its advantages in handling optimization problems with discrete or continuous variables, as well as its simplicity in implementation and parameter tuning. However the choice of algorithm should be aligned with the specific problem and the desired performance criteria; in some cases, other evolutionary algorithms, such as Genetic Algorithms (GA), may outperform PSO.

2. Materials and Methods

2.1. Dota 2[4] with All-Pick Mode

As noted earlier, Dota 2 is a highly popular MOBA game with a complex gameplay system offering a variety of modes[5], such as All Pick, Turbo Mode, Ranked All Pick, New Player Mode, Play Vs Bots, 1 vs 1 Solo Mid, and more. Since this research focuses on enhancing team strategy in Dota 2 using evolutionary algorithms, the key components that will be utilized are: (1) All Pick Mode; (2) The Heroes that have been used are Sniper, Gyrocopter, Wraith King, Necropos, and Lion; (3) The selected item will be fixed according to the Heroes used; (4) The selected skill will also be fixed according to the Heroes used; (5) This research will improve team strategy by controlling the team's desire. To better understand the research methodology used in this study, it is essential to have a deeper understanding of Dota 2's gameplay mechanism, programming languages, file structures, and methods for incorporating Artificial Intelligence into the game.

2.2. Gameplay All Pick Mode [25]

The All Pick mode, one of the most commonly used modes in Dota 2, will be utilized in this research to test the team's strategies. This mode is ideal for strategy, as it allows players to select any Hero available in the game. The rules for All Pick Mode include: (1) Players can choose any Hero; (2) 75 seconds for Hero selection time; (3) 75 seconds for pre-creep time; (4) The player loses 2 gold per second if they didn't pick any Heroes after time out; (5) The first team that destroys the 'Throne' of the enemy will win; (6) Players can choose the same heroes. Apart from the 6 special rules above, how to play this mode is shown in Figure 1.

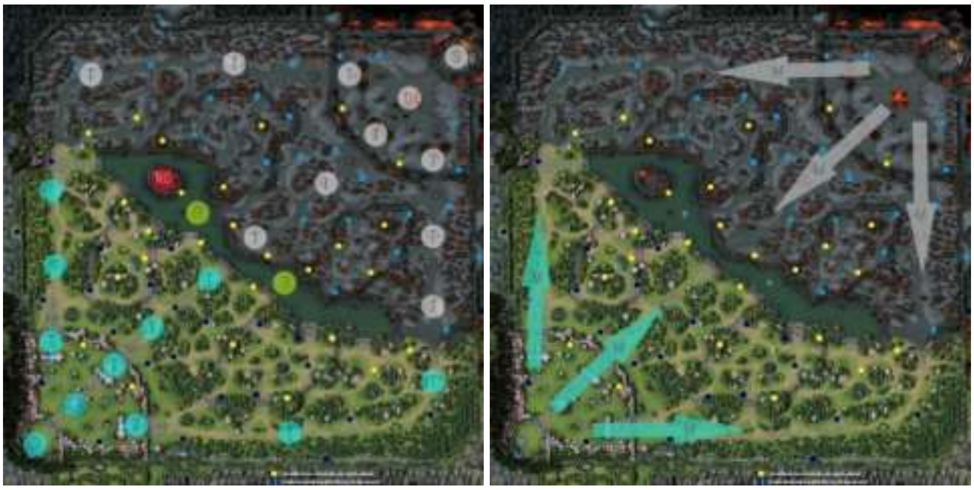
In Figure 1, the color Cyan represents the Radiant team, while White represents the Dire team. The game begins at the initial position (S) from which Heroes can move in any direction on the map. To achieve the main objectives of this mode, each Hero must defend or attack the tower (T) in their respective lanes. Every 30 seconds, Creeps (NPC) spawn in all lanes and advance toward the enemy's area. Heroes gain levels by defeating enemy Creeps or Heroes, which strengthens them. Additionally, Heroes can grow stronger by collecting Runes (R) or defeating Roshan (RO) to support an attack on a random lane. Runes appear every 2 minutes at random locations, with 8 types in Dota 2: Double Damage, Haste, Illusion, Regeneration, Invisibility, Arcane, Bounty, and Water. The game concludes when one team destroys the opposing team's Throne (TH).

---

[3] These two concepts are important for evolutionary algorithms because they can support each other between searching for old values and new values.
[4] This study uses Dota 2 version 7.3.2. For newer versions, further research should be carried out.
[5] Taken from the Dota 2 website: https://dota2.fandom.com/wiki/Game_modes

http://doi.org/10.26594/register.v10i2.3503

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
10 (2) July 2024 116-126



(a) (b)

Fig. 1. Dota 2 Map: (a) Object Marking and (b) Game Movement

2.3. Dota 2 Artificial Intelligence Scripting[6]

The scripting language used in Dota 2 is Lua [26], which has full access to the game's API. The API provides information on game state conditions, unit locations, skill cooldown values, and other data relevant to gameplay. However, the API includes limitations to prevent cheating, such as restricting control over creep units or other player's units. In Dota 2, there are three levels of evaluation and decision-making: Team Level, Mode Level, and Action Level. The Team Level guides each agent to move in alignment with the designated strategy, while the Mode and Action Levels are decisions made by each agent during specific game activities, like laning, attacking, shopping, and more. Table 1 lists Dota 2 activity groups by level.

Table 1. Dota 2 AI Level

Team Level	**Mode Level**	**Action Level**
Push Each Lane	Laning	Moving to a Location
Defend Each Lane	Attack	Attacking a Unit
Farm Each Lane	Roam	Using an Ability
Roam	Retreat	Purchasing an Item
Kill Roshan	Secret Shop	Etc.
	Side Shop	
	Rune	
	Push Tower Top	
	Push Tower Mid	
	Push Tower Bot	
	Defend Tower Top	
	Defend Tower Mid	
	Defend Tower Bot	
	Assemble	
	Team Roam	
	Farm	
	Defend Ally	
	Evasive Maneuvers	
	Roshan	
	Item	
	Ward	

For bot scripts to be uploaded and executed on the Dota 2 server, they must be placed in the game/dota/scripts/vscripts/bots folder. This step is essential in the development process as the server recognizes AI scripts only from this specific folder. This setup helps prevent cheating and ensures that the AI runs through the Dota 2 server. Additionally, the API for bot script development is designed to allow for easy modification of various elements, enabling seamless selection and management of functions and files. Table 2 outlines the steps for overwriting the Dota 2 AI script.

[6] Taken from Dota 2 Developer Guide: https://developer.valvesoftware.com/wiki/Dota_Bot_Scripting

http://doi.org/10.26594/register.v10i2.3503

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
10 (2) July 2024 116-126

Table 2. Dota 2 Code Overwrite

Scope	Overwrite File Name	Overwrite Function
Complete Takeover	bot_[Hero Name].lua (Example: bot_lina.lua), if you want to overwrite all bots then change Hero Name with generic	Think()
Mode Overwrite	mode_[Mode Name]_[Hero Name].lua (Example: mode_laning_lina.lua), if you want to overwrite all bots then change Hero Name with generic	GetDesire() OnStart() OnEnd() Think()
Ability and Item Usage	ability_item_usage_[Hero Name].lua (Example: ability_item_usage_lina.lua), if you want to overwrite all bots then change Hero Name with generic	ItemUsageThink() AbilityUsageThink() CourierUsageThink() BuybackUsageThink() AbilityLevelUpThink()
Minion Control	item_purchase_[Hero Name].lua (Example: item_purchase_lina.lua), if you want to overwrite all bots then change Hero Name with generic	MinionThink(hMinionUnit)
Team Level Desires	team_desires.lua	TeamThink() UpdatePushLaneDesires() UpdateDefendLaneDesires() UpdateFarmLaneDesires() UpdateRoamDesire() UpdateRoshanDesire()
Hero Selection	hero_selection.lua	Think() UpdateLaneAssignments() GetBotNames()

In this research, the team-level desires file will be overwritten to manage the five components of the team level, as listed in Table 1. Each component has a specific function and value, detailed in Table 3, with each function being called every frame. Typically, the return value of these functions is fixed or rule-based. However, this study will use Particle Swarm Optimization to determine the optimal return value for each frame. By applying an evolutionary algorithm, more dynamic values can be assigned to team desires, allowing the AI to evaluate its current conditions in each frame before selecting the best-desired values to apply.

Table 3. Dota 2 Team Level

Team Level	Function Name	Function Return Value
Push Each Lane	UpdatePushLaneDesires()	3 floating values between 0 and 1, represent the top lane, middle lane, and bottom lane
Defend Each Lane	UpdateDefendLaneDesires()	3 floating values between 0 and 1, represent the top lane, middle lane, and bottom lane
Farm Each Lane	UpdateFarmLaneDesires()	3 floating values between 0 and 1, represent the top lane, middle lane, and bottom lane
Roam	UpdateRoamDesire()	Floating values between 0 and 1 and Heroes that become roam target
Kill Roshan	UpdateRoshanDesire()	Floating values between 0 and 1, represent the desire to kill Roshan

2.4. Particle Swarm Optimization [27]–[30]

Particle Swarm Optimization is one of the earliest evolutionary algorithms. It was the first to introduce a floating-point representation, allowing for faster processing of candidate solutions compared to permutation representations. This floating-point representation is particularly suitable for this study, as all data that needs to be modified and managed regarding the team's desire consists of float values. Figure 2 illustrates the Particle Swarm Optimization algorithm used in this research, which has been adapted for the implementation of team's desired in Dota 2.

 ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
10 (2) July 2024 116-126

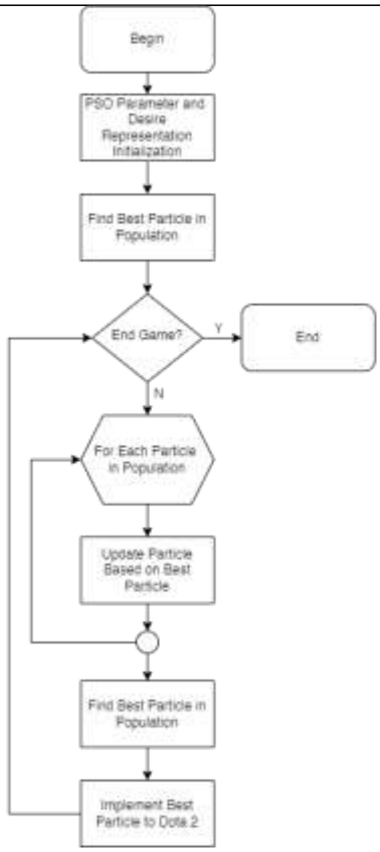


Fig. 2. Particle Swarm Optimization Algorithm

For each particle, updates are performed using formulas (1) and (2). Formula (1) adjusts the particle's velocity, while formula (2) updates the particle's value, commonly referred to as the particle's position. Particles will continue to be updated for each game frame until the game concludes. The best value generated by PSO will be applied as the desired value for the team in that frame.

$$V_i = w * V_i + C_p * R_p * (P_i - X_i) + C_g * R_g * (G_i - X_i) \quad (1)$$

$$X_i = X_i + V_i \quad (2)$$

Where X = position, V = velocity, w = inertia weight, Cp = constant for particle best, Cg = constant for global best, Rp and Rg = random value between 0 and 1, P = best value for each particle, and G = best particle in this population.

2.5. Dota 2 Team's Desired Representation

In this study, the desired team strategy is represented by a vector consisting of 12 elements: 3 for push lanes, 3 for defending lanes, 3 for farming lanes, 2 for roaming, and 1 for killing Roshan. Each element corresponds to a specific desire within the Dota 2 game. The 3 values for the push lane, defend lane, and farm lane correspond to the top lane, middle lane, and bottom lane, respectively. The 2 values for roaming indicate the team's desire to roam and the targeted heroes for ganking. The single value for killing Roshan reflects the team's intention to attack and eliminate Roshan. Figure 3 illustrates this representation.

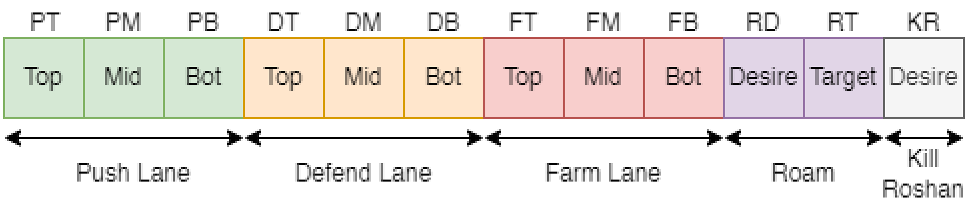


Fig. 3. Team's Desired Representation

Each element in the vector representation will have a value ranging from 0 and 1. However, after the initialization process or when updating the particle values, the following post-processing steps will be carried out: (1) Clipping Value, if the value generated by PSO is below 0 or above 1, clipping is carried out according to formula $X_i = \min(\max(X_i, 0), 1)$ so that the range of particle values does not change;

 http://doi.org/10.26594/register.v10i2.3503

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
10 (2) July 2024 116-126

(2) Normalize Value, In particular, additional processing is required for lane representation, as it is common for the Push, Defend, and Farm strategies to have the same value. These three strategies have different purposes: Push is used to attack towers, Defend is for protecting towers, and Farms is aimed at maximizing gold and experience gained in the lane area. Given these differences, this study normalizes the values for the push, defend, and farm strategies in each lane so that their total equals 1. Formula 4 is applied to normalize the values of the 3 strategies in each lane.

$$PL = \frac{PL}{PL+DL+FL},$$
$$DL = \frac{DL}{PL+DL+FL}, and$$
$$FL = \frac{FL}{PL+DL+FL} \quad (4)$$

Where L is the representation of the calculated Lane. For example, if you want to calculate the Top Lane, replace all L with T. So the values used are PT, DT, and FT.

At the end of the iteration, the best particle will be selected and applied to the Team Desired values. There are 2 types of implementation: the first is to apply the values directly, and the values for lanes and killing Roshan can be used without modification since they share the same range (between 0 and 1). The second type involves preprocessing, as the Roam value cannot be used directly because it is intended for a single Hero. Therefore, using Table 4, the RT Value is utilized to select the Hero designated for Roaming (RD).

Table 4. Distribution of RT Values

RT Range	Hero ID
0.00 – 0.20	1
0.21 – 0.40	2
0.41 – 0.60	3
0.61 – 0.80	4
0.81 – 1.00	5

2.6. Dota 2 Fitness Function

In this research, the fitness function used is based on the team's overall net worth, which can accessed by the AI through the DOTA 2 API. The team's net worth serves as a strong indicator of its performance; a higher net worth typically suggests a stronger team. The fitness function is calculated for every frame using the team's net worth and is employed to evaluate the performance of each particle. Formula 5 represents the fitness function used in this study.

$$F(X) = \sum SL * FSL(X) + RD * FRD(X) + KR * FKR(X) \quad (5)$$

In this context, F(X) represents the fitness function, X denotes a particle, S refers to the strategy used (Push, Defend, or Farm), L indicates the lane being calculated (Top, Middle, Bottom), and F is the fitness function for each strategy. For each Push strategy, whether on the Top Lane (FPT), Middle Lane (FPM), or Bottom Lane (FPB), formula 6 is employed for the fitness calculation.

$$FPL(X) = (3 - countEnemyTowerL) + (5 - countNearEnemyHero) + (5 - countLiveEnemyHero) + (countLiveTeamHero) \quad (6)$$

In this context, countEnemyTower represents the number of enemy towers that still exist in lane L, countNearEnemyHero denotes the number of enemy heroes near our AI, countLiveEnemyHero indicates the number of enemy heroes that are still alive, and countLiveTeamHero refers to the number of our heroes that still alive. For each Defend strategy, whether on the Top Lane (FDT), Middle Lane (FDM), or Bottom Lane (FDB), formula 7 is used for the fitness calculation.

$$FDL(X) = (3 - countTeamTowerL) + (countNearEnemyHero) + (countLiveEnemyHero) + (5 - countLiveTeamHero) \quad (7)$$

In this context, countTeamTower represents the number of team towers that still exist in lane L. Last in strategy in lane is Farm strategy, for each of it whether on the Top Lane (FFT), Middle Lane (FFM), or Bottom Lane (FFB), formula 8 is used for the fitness calculation.

$$FFL(X) = (5 - countNearEnemyHero) + (20 - heroLevel) \quad (8)$$

In this context, heroLevel refers to the current level of the hero controlled by the AI. The maximum hero level in Dota 2 is 30; however, this fitness function sets the maximum to 20[7], as this level is considered optimal for prioritizing other strategies over farming strategies. The remaining 10

[7] Based on discussions with the Dota 2 community and trials conducted for max levels 20, 25 and 30. The best is at number 20.

http://doi.org/10.26594/register.v10i2.3503

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
10 (2) July 2024 116-126

levels, will increase gradually as the game progresses. For the Roam strategy, formula 9 is used as the fitness function.

$$FRD(X) = \sum(1 - countNearEnemyOtherHero) + \sum(countHeroInLane) \quad (9)$$

In this context, countNearEnemyOtherHero represents the number of enemy heroes surrounding our team. The fitness function takes into account the number of enemy heroes near our team and the number of heroes present in each lane. When only one enemy hero is nearby, the impact of the Roam strategy is greater compared to situations where multiple enemy heroes are present. Additionally, countHeroInLane is calculated to determine the number of heroes in each lane, which is crucial since each lane should have at least one hero to defend it. If no hero is guarding a lane, the Roam strategy should not be utilized, as it may result in enemy attacks on that lane. Finally, for the Kill Roshan strategy, formula 10 is used as a fitness function.

$$FKR(X) = \left(isRoshanAvailable\ ?\ 4^{8} : 0\right) * (5 - countLiveEnemyHero) + countLiveTeamHero \quad (10)$$

In this context, isRoshanAvailable indicates whether Roshan has appeared in the game. If Roshan is present, he is assigned a weight of 4 because he is a game character capable of significantly altering the course of the match. This weight assignment is also influenced by the number of enemy players and team players who are still alive. If too many enemy players are alive, the team may struggle to secure a victory; conversely, if there are too few team players, the team may not be able to successfully kill Roshan.

3. Results and Discussion

Four trials were conducted in this study: PSO vs GA, AI vs AI, AI vs Player, and expert judgment. For each trial, a minimum of 3 matches was played to ensure that the results were valid and not simply due to luck[9]. The trials utilized three difficulty levels: Easy, Normal, and Hard. Although Dota 2 offers an Unfair level, it was not included in this study, as this level provides excessive advantages that can negatively impact the player's experience. The sections below will provide a more detailed explanation of each trial conducted.

3.1. PSO vs Genetic Algorithm (GA)

This trial involved a comparison of the algorithm used for enhancement. The GA algorithm [31]–[33] was chosen for comparison due to it is status as a classic evolutionary algorithm that remains widely applicable in various scenarios. Table 5 shows the results of the matches played by the two AIs.

Table 5. PSO Enhanced AI Vs GA Enhanced AI Results[10]

	Easy (GA)	**Normal (GA)**	**Hard (GA)**
Easy (PSO)	4 Win 0 Lose		
Normal (PSO)		3 Win 1 Lose	
Hard (PSO)			2 Win 2 Lose

Each match in Table 5 was played four times, with both enhanced AIs using the same hero composition to ensure balanced gameplay. The results in Table 5 indicate that in easier matches, PSO outperforms GA, while their abilities are approximately equal in more difficult matches. This discrepancy can be attributed to PSO's faster processing speed, which enables the AI to make quicker decisions. However, this speed advantage does not significantly enhance performance when the AI must prioritize not only rapid decisions but also high-quality ones.

3.2. Enhanced AI Vs Default AI Trial

The trial involved a comparison between the Dota 2 AI-enhanced by the team's desires and the default Dota 2 AI. This comparison was made to emphasize the research's focus on team desires, while all other

[8] Trial is done for numbers 1 to 5. However, 4 is the best value. Numbers 1 to 3, do not trigger the AI's desire to attack Roshan even though the opponent's hero is dead. While the number 5, makes AI's desire to attack Roshan too domineering. Even though only 1 opponent's hero died, AI still attacked Roshan and made it easy for the opponent to take Roshan from AI's hands.

[9] Because the initial value of the evolutionary algorithm is a random value and the composition of the opponent's players is also not fixed if you only do one tryout, you will appear lucky either because of the random value or because your opponent chooses a hero that is easy to beat.

[10] We only do Easy (PSO) Vs Easy (GA), Normal (PSO) Vs Normal (GA), and Hard (PSO) Vs Hard (GA) because algorithm comparisons must be conducted under the same conditions to accurately assess the superiority of one algorithm over the other.

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
10 (2) July 2024 116-126

AI functions remained unchanged from the defaults provided by Dota 2. Table 6 shows the results of the matches played between the two AIs.

Table 6. Enhanced AI Vs Default AI Results[11]

	Easy (Default)	Normal (Default)	Hard (Default)
Easy (Enhanced)	3 Win 0 Lose	1 Win 2 Lose	0 Win 3 Lose
Normal (Enhanced)		2 Win 1 Lose	0 Win 3 Lose
Hard (Enhanced)			2 Win 1 Lose

For each match in Table 6, either easy vs easy or another matchup was played 3 times. In every match, Enhanced AI consistently used the same Hero composition, while the Default AI selected Heroes based on its own preferences. The results presented in Table 6 reflect the performance of the Enhanced AI. It is evident that incorporating team desires using PSO improves AI performance; however, it still struggles to compete effectively at different difficulty levels, such as Easy vs Normal or Easy vs Hard. This difficulty arises from the variations in decision-making speeds and action sets required at different levels.

3.3. Enhanced AI Vs Player Trial

The trial involved 30 players from the Dota 2 community, who were grouped into 3 categories based on their MMR[12]. The groups were organized as follows: players with an MMR below 1000 (10 players divided into 2 teams), players with an MMR between 1000 to 3000 (10 players divided into 2 teams), and players with an MMR above 3000 (10 players divided into 2 teams). This categorization was based on skill level, with MMR below 1000 representing beginners and MMR above 3000 indicating experienced and reliable players. Table 7 is the result of the match between Enhanced AI and players.

Table 7. Enhanced AI Vs Player Results[13]

	Below 1000		1000 – 3000		Above 3000	
	Team 1	**Team 2**	**Team 1**	**Team 2**	**Team 1**	**Team 2**
Easy (Enhanced)	2 Win 1 Lose	1 Win 2 Lose				
Normal (Enhanced)	3 Win 0 Lose	2 Win 1 Lose	1 Win 2 Lose	1 Win 2 Lose		
Hard (Enhanced)	3 Win 0 Lose	3 Win 0 Lose	2 Win 1 Lose	3 Win 0 Lose	0 Win 3 Lose	1 Win 2 Lose

As with previous matches, each match was held three times, with the Enhanced AI using the same Hero composition while players were free to choose heroes according to their preferences. The results in Table 7 reflect the performance of the Enhanced AI. It is evident that while the Enhanced AI can defeat players with low MMR, it struggles against those with high MMR. Analyzing the team's desire, the Enhanced AI appears to provide appropriate strategies; however, high MMR players are often unpredictable and rarely visible on the screen. Consequently, the fitness function, which accounts for the number of opposing players, becomes less impactful, leading the AI to underestimate its opponents since they are not frequently seen. Players typically engage only when attempting to kill AI or attack towers.

Of the 30 players who participated, three were selected to provide feedback on the Enhanced AI. These players are experts in using multiple heroes and have MMRs above 3000 (3200, 3500, and 3800). Their feedback is based on the 36 matches played. They noted that the enhanced AI performs better than the default AI in Dota 2, as it can dynamically choose its playing strategies (push, defend, farm, roam, and kill Roshan), making its movements more difficult to predict. In contrast, the Default AI tends to be more monotonous in its strategy choices. However, there are two drawbacks to Enhanced AI. First, when a player disappears, the AI tends to default to the Push or Farm strategy, which can make it

[11] We didn't do Normal (Enahned) Vs Easy (Default), Hard (Enhanced) vs Easy (Default), and Hard (Enhanced) vs Normal (Default) because it is normal to win. The higher level is better than the lower level.

[12] MMR is Matchmaking Rating. Dota 2 uses MMR to determine the skills of players and grouping during matchmaking so that players will face other players who have a similar MMR. The higher the skill, the higher the MMR. Currently, the MMR range in Dota 2 is 1 to 13000. However, this study limits it to only 3000 because that number is the lower limit for a reliable player.

[13] We didin't do Easy (Enhanced) vs Above 1000 MMR and Normal (Enhanced) vs Above 3000 MMR because it will never win. Without change the decision making that AI used, it will imposible to overcome the differences in skill although the desired is better.

http://doi.org/10.26594/register.v10i2.3503

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
10 (2) July 2024 116-126

vulnerable to being easily attacked and having towers destroyed in that lane. Second, while the AI's dynamic strategy selection is an improvement, it is not supported by strong decision-making. As a result, even when the strategy is appropriate, the decisions made can still be predictable and easy to counter. To develop a truly effective AI, it is recommended that the decision-making function also be enhanced.

4. Conclusion

The conclusion drawn from the trials conducted in this study is Particle Swarm Optimization can enhance the performance of Team Desire in Dota 2 AI. While these improvements allow the enhanced AI to compete effectively against the Default AI at the same level or against players with low MMR, it still struggles to defeat higher-level opponents. Therefore, future research should focus on two key areas: first, modifying or adding to the fitness function to better support scenarios where opponents disappear from the screen (jungling); and second, improving the decision-making function to ensure that the strategies employed by the AI are fully optimized.

Author Contributions

H. Armanto: Conceptualization, formal analysis, methodology, project administration, resources, software, validation, and writing – original draft. H. A. Rosyid: Supervision and writing – review & editing. M. Muladi: Supervision and writing – review & editing. G. Gunawan: Supervision and writing – review & editing.

Declaration of Competing Interest

We declare that we have no conflict of interest.

References

[1] Z. Hellman and M. Pintér, "Charges and bets: a general characterisation of common priors," *Int. J. Game Theory*, vol. 51, no. 3, pp. 567–587, 2022, doi: 10.1007/s00182-022-00805-4.

[2] T. Hazra and K. Anjaria, "Applications of game theory in deep learning: a survey," *Multimed. Tools Appl.*, vol. 81, no. 6, pp. 8963–8994, 2022, doi: 10.1007/s11042-022-12153-2.

[3] E. R. Sadik-Zada, A. Gatto, L. Aldieri, G. Bimonte, L. Senatore, and C. P. Vinci, "Game Theory Applications to Socio-Environmental Studies, Development Economics, and Sustainability Research," *Games*, vol. 15, no. 1. 2024, doi: 10.3390/g15010005.

[4] D. Bulut, Y. Samur, and Z. Cömert, "The effect of educational game design process on students' creativity," *Smart Learn. Environ.*, vol. 9, no. 1, p. 8, 2022, doi: 10.1186/s40561-022-00188-9.

[5] Y. Gui, Z. Cai, Y. Yang, L. Kong, X. Fan, and R. H. Tai, "Effectiveness of digital educational game and game design in STEM learning: a meta-analytic review," *Int. J. STEM Educ.*, vol. 10, no. 1, p. 36, 2023, doi: 10.1186/s40594-023-00424-9.

[6] P. W. Atmaja, D. O. Siahaan, and I. Kuswardayan, "Game Design Document Format For Video Games With Passive Dynamic Difficulty Adjustment," *Regist. J. Ilm. Teknol. Sist. Inf.*, vol. 2, no. 2 SE-Article, pp. 86–97, Jul. 2016, doi: 10.26594/register.v2i2.551.

[7] B. Xia, X. Ye, and A. Abuassba, *Recent Research on AI in Games*. 2020.

[8] N. A. Barriga, M. Stanescu, F. Besoain, and M. Buro, "Improving RTS Game AI by Supervised Policy Learning, Tactical Search, and Deep Reinforcement Learning," *IEEE Comput. Intell. Mag.*, vol. 14, no. 3, pp. 8–18, 2019, doi: 10.1109/MCI.2019.2919363.

[9] S. Karim, "Perubahan perilaku Non-Player Character (NPC) pada Game Arabic Hunter menggunakan Jaringan Syaraf Tiruan Perceptron," *Regist. J. Ilm. Teknol. Sist. Inf.*, vol. 3, no. 1 SE-Article, pp. 34–41, Jan. 2017, doi: 10.26594/register.v3i1.622.

[10] H. Armanto, H. A. Rosyid, Muladi, and Gunawan, "Improved Non-Player Character (NPC) behavior using evolutionary algorithm—A systematic review," *Entertain. Comput.*, vol. 52, p. 100875, 2025, doi: https://doi.org/10.1016/j.entcom.2024.100875.

[11] M. Mora-Cantallops and M.-Á. Sicilia, "MOBA games: A literature review," *Entertain. Comput.*, vol. 26, Feb. 2018, doi: 10.1016/j.entcom.2018.02.005.

[12] G. Robertson and I. Watson, "A Review of Real-Time Strategy Game AI," *AI Mag.*, vol. 35, no. 4, pp. 75–104, 2014, doi: 10.1609/aimag.v35i4.2478.

[13] D. Mauleon, *Online Battle Arena Esports: The Competitive Gaming World of League of Legends, Dota 2, and More!* Capstone, 2019.

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
10 (2) July 2024 116-126
[14] E. F. Fangasadha, S. Soeroredjo, Anderies, and A. A. S. Gunawan, “Literature Review of OpenAI Five’s Mechanisms in Dota 2’s Bot Player,” in *2022 International Seminar on Application for Technology of Information and Communication (iSemantic)*, 2022, pp. 183–190, doi: 10.1109/iSemantic55962.2022.9920480.
[15] C. Berner *et al.*, “Dota 2 with large scale deep reinforcement learning,” *arXiv Prepr. arXiv1912.06680*, 2019.
[16] A. Semenov, P. Romov, S. Korolev, D. Yashkov, and K. Neklyudov, “Performance of Machine Learning Algorithms in Predicting Game Outcome from Drafts in Dota 2,” 2016.
[17] C. H. Ke *et al.*, “DOTA 2 match prediction through deep learning team fight models,” in *2022 IEEE Conference on Games (CoG)*, 2022, pp. 96–103, doi: 10.1109/CoG51982.2022.9893647.
[18] D. Freitas, L. G. Lopes, and F. Morgado-Dias, “Particle Swarm Optimisation: A Historical Review Up to the Current Developments,” *Entropy*, vol. 22, no. 3. 2020, doi: 10.3390/e22030362.
[19] Q. Xiong, X. Zhang, X. Xu, and S. He, “A Modified Chaotic Binary Particle Swarm Optimization Scheme and Its Application in Face-Iris Multimodal Biometric Identification,” *Electronics*, vol. 10, no. 2. 2021, doi: 10.3390/electronics10020217.
[20] S. Pervaiz, Z. Ul-Qayyum, W. H. Bangyal, L. Gao, and J. Ahmad, “A Systematic Literature Review on Particle Swarm Optimization Techniques for Medical Diseases Detection,” *Comput. Math. Methods Med.*, vol. 2021, p. 5990999, 2021, doi: 10.1155/2021/5990999.
[21] Y. Yang, Q. Liao, J. Wang, and Y. Wang, “Application of multi-objective particle swarm optimization based on short-term memory and K-means clustering in multi-modal multi-objective optimization,” *Eng. Appl. Artif. Intell.*, vol. 112, p. 104866, 2022, doi: https://doi.org/10.1016/j.engappai.2022.104866.
[22] C. Zhang, H. Shao, and Y. Li, “Particle swarm optimisation for evolving artificial neural network,” in *Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics. “cybernetics evolving to systems, humans, organizations, and their complex interactions” (cat. no.0*, 2000, vol. 4, pp. 2487–2490 vol.4, doi: 10.1109/ICSMC.2000.884366.
[23] S. Mirjalili, J. Song Dong, A. Lewis, and A. S. Sadiq, “Particle Swarm Optimization: Theory, Literature Review, and Application in Airfoil Design BT - Nature-Inspired Optimizers: Theories, Literature Reviews and Applications,” S. Mirjalili, J. Song Dong, and A. Lewis, Eds. Cham: Springer International Publishing, 2020, pp. 167–184.
[24] H. Armanto, H. A. Rosyid, and others, “Evolutionary Algorithm in Game--A Systematic Review,” *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, 2023.
[25] I. Media, *A Beginner’s Guide to DOTA 2*. CreateSpace Independent Publishing Platform, 2014.
[26] R. Ierusalimschy, *Programming in Lua*. Roberto Ierusalimschy, 2006.
[27] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95 - International Conference on Neural Networks*, 1995, vol. 4, pp. 1942–1948 vol.4, doi: 10.1109/ICNN.1995.488968.
[28] Y. Shi and R. Eberhart, “A modified particle swarm optimizer,” in *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, 1998, pp. 69–73, doi: 10.1109/ICEC.1998.699146.
[29] R. C. Eberhart and Y. Shi, “Comparing inertia weights and constriction factors in particle swarm optimization,” in *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)*, 2000, vol. 1, pp. 84–88 vol.1, doi: 10.1109/CEC.2000.870279.
[30] M.-P. Song and G.-C. Gu, “Research on particle swarm optimization: a review,” in *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.04EX826)*, 2004, vol. 4, pp. 2236–2241 vol.4, doi: 10.1109/ICMLC.2004.1382171.
[31] S. Sendari, A. N. Afandi, I. A. E. Zaeni, Y. D. Mahandi, K. Hirasawa, and H.-I. Lin, “Exploration of genetic network programming with two-stage reinforcement learning for mobile robot,” *TELKOMNIKA (Telecommunication Comput. Electron. Control.*, vol. 17, no. 3, pp. 1447–1454, 2019.
[32] D. Lestari, S. Sendari, and I. A. E. Zaeni, “Genetic algorithm for finding shortest path of mobile robot in various static environments,” *J. Infotel*, vol. 15, no. 3, pp. 280–286, 2023.
[33] S. Sendari, A. B. Putra Utama, N. S. Fanany Putri, P. Widiharso, and R. J. Putra, “K-Means and Fuzzy C-Means Optimization using Genetic Algorithm for Clustering Questions ,” *Int. J. Adv. Sci.*

ISSN 2502-3357 (online) | ISSN 2503-0477 (print)
10 (2) July 2024 116-126

Comput. Appl., vol. 1, no. 1 SE-Articles, pp. 1–9, Dec. 2021, doi: 10.47679/ijasca.v1i1.2.

[34] M. Wu, S. Xiong, and H. Iida, *Fairness Mechanism in Multiplayer Online Battle Arena Games*. 2016.