

Summary:

If any of the AI's pokemon have a move that is SE on the player's pokemon, pick the pokemon that has the worst defensive typing for the player's current pokemon. If none of their mons have a SE move pick the pokemon in the AI's party with the move with the best type effectiveness, if the just fainted pokemon had STAB with a move prioritize that one. All ties go to the first pokemon in the party of the tied mons. Status moves count for type effectiveness for the second step, but not the initial step.

Double battles appear to work exactly the same as single battles, except at the start of the switching logic it randomly picks one pokemon on the player's side to base everything off. Even partner AI in double battles base switching logic off the players side as far as we can tell.

The nitty gritty is below.

The switching logic has two sections, and the second is used only if the first fails to provide any valid switches.

Universal Information

The AI will account for any type changes the player is under, but will not account for type changes of its own pokemon.

The AI will see Normal and Fighting typed moves as being neutrally effective on the player's Ghost types, and will not care if you used Foresight on it's just fainted pokemon.

All ties go to the first pokemon in the party of the tied mons.

Section 1: "Best" Type Matchup (with a super effective move)

Give each pokemon a score, and pick the pokemon with the highest score. Score is calculated as follows

The game starts with a value of 10, if a type matchup is an immunity it multiplies it by 0, if it is not very effective it halves it, and if it is super effective it doubles it. It rounds down after every check, which can lead to discrepancies.

The type checks are done first for the primary typing of the player, then the secondary typing of the player. These are calculated with the player as the attacker and the AI as the defender, against a dual type pokemon the order the type effectiveness of the AI's defending pokemon is hard coded, refer to the end of this document for the complete list. Abilities are ignored.

A single typed pokemon on the player side applies the resistances or weaknesses of the AI's defender twice here, as internally single typed pokemon are stored as dual types with the same type twice. This does not apply to the defender, only the attacker.

In case of two pokemon with the same score, pick the one that was found first. Once you have determined the pokemon with the highest score, check if that pokemon has a move that is super effective on the players pokemon. Status moves are never counted as super effective in this step.

If the pokemon had a super effective move, switch it in, otherwise check for a super effective move on the next highest score until every pokemon has been checked. The AI has knowledge that a ground type move is never super effective on a pokemon with levitate at this stage, but no other knowledge of abilities. A pokemon with a score of 0 is never chosen in this section, this is done by having an immunity to either one of the attackers types, or by the AI having a dual type pokemon where both types resist all of your types.

Examples:

Example 1: Rounding down

Player has a combusken out, AI has a corsola and a yanma. All of the AI's pokemon have a move that is SE on combusken.

Fire vs Water: $10 * 0.5 = 5$

Fire vs Rock: $5 * 0.5 = 2$ (Rounded down)

Fighting vs Rock: $2 * 2 = 4$

Corsola has a score of 4

Fire vs Bug: $10 * 2 = 20$

Fighting vs Flying: $20 * 0.5 = 10$

Fighting vs Bug: $10 * 0.5 = 5$

Yanma has a score of 5

As yanma's score is higher than corsola's, yanma will always be picked over Corsola.

Example 2: Lack of SE move

Player has a combusken out, AI has a rhyhorn, a yanma, and a magnemite. Rhyhorn and yanma have SE moves on combusken, but magnemite does not.

Fire vs Rock: $10 * 0.5 = 5$

Fighting vs Rock: $5 * 2 = 10$

Rhyhorn has a score of 10

Fire vs Bug: $10 * 2 = 20$

Fighting vs Flying: $20 * 0.5 = 10$

Fighting vs Bug: $10 * 0.5 = 5$

Yanma has a score of 5

Fire vs Steel: $10 * 2 = 20$

Fighting vs Steel: $20 * 2 = 40$

Magnemite has a score of 40

Magnemite has the highest score, but as it has no SE moves it won't be chosen. Rhyhorn has a higher score than Yanma and both have SE moves, so Rhyhorn will be chosen.

Example 3: Type matchup ordering

Player has a combusken out, AI has a lunatone, and a yanma. Both have SE moves on combusken.

Fire vs Bug: $10 * 2 = 20$

Fighting vs Flying: $20 * 0.5 = 10$

Fighting vs Bug: $10 * 0.5 = 5$

Yanma has a score of 5

Fire vs Rock: $10 * 0.5 = 5$

Fighting vs Psychic: $5 * 0.5 = 2$ (rounded down)

Fighting vs Rock: $2 * 2 = 4$

Lunatone has a score of 4

Yanma has the highest score here because of the internal ordering of the type matchups causing it to resolve fighting vs psychic before it resolves fighting vs rock despite rock being the primary typing of Lunatone.

Example 4: Single typed attackers and score 0

Player has a typhlosion out, AI has a charizard and a corsola. The corsola has a SE move but the charizard does not.

Fire vs Fire: $10 * 0.5 = 5$

Fire vs Fire: $5 * 0.5 = 2$ (rounded down)

Charizard has a score of 2

Fire vs Water: $10 * 0.5 = 5$

Fire vs Rock: $5 * 0.5 = 2$ (rounded down)

Fire vs Water: $2 * 0.5 = 1$

Fire vs Rock: $1 * 0.5 = 0$ (rounded down)

Corsola has a score of 0

Despite the charizard having a higher score, since it lacks a SE move it will not be chosen. Normally this would result in corsola being chosen as it has a SE move, but as corsola has a score of 0 it can never be chosen so it would move on to section two.

Section 2: "Highest Damage Move"

The AI calculates the damage it would do with the most recently used move with its fainted pokemon upon the target pokemon, this is regardless of whether or not the player or the AI used the most recent move. It always uses max damage, and never crits, so this is unaffected by those elements of randomness. At this point, it does see things such as rain, choice bands, and

most effects which modify damage, it does not see type effectiveness, STAB, or immunity granting abilities. Status moves are treated as doing 3 damage.

Then it runs down the list of pokemon in order, for each one applying STAB and type effectiveness, this allows for status moves to be treated as damaging and the AI will use the most recently fainted pokemon's typing to determine if the move should have STAB. The AI will ignore all non-status moves with non-standard damage in this section, a full list of these moves can be seen below the examples.

Note that type effectiveness is applied in the order listed in the type interactions section, which can cause rounding errors due to integer arithmetic if the initial damage was odd and the player's pokemon has two types which resist and are weak to two different types. See example 4.

The AI is looking for its highest damage move, it starts with a best seen damage value of 0. When it's running down the list, if the move is calculated to do more than the previous best seen damage it remembers the pokemon, stores the damage the move did as the new best seen damage and continues. However, when storing the damage it stores it modulo 256. So for example, if the move would do 330 damage, it will think 330 damage is high so that is a good move, but when storing it would only store it as doing 74 damage, or $330 - 256$. If it was calculated to be 660 it would think it is very good again, but it would only store 148, or $660 - 256 - 256$. This would cause the next move over the stored value to get treated as the highest damage move, such as an 80 damage move after 74 had been stored.

If you have a pokemon with Levitate out, the AI will think ground moves deal neutral damage against it in this section, regardless of typing. The AI will not see Wonder Guard or Thick Fat in this step, it is believed that the AI will not see Flash Fire, Soundproof, Volt Absorb, or Water Absorb in this section.

In the very rare circumstance that your pokemon is immune to every move that the AI has on all of their still living pokemon, including their status moves based off typing, the AI should just pick whatever pokemon is next in the party.

Example 1: The player has a water type pokemon out, the AI has a pokemon with Tackle, and a pokemon with Thunderbolt, it will pick the latter because it is super effective.

Example 2: The player has a water type pokemon out, the AI has a pokemon with Tackle and a pokemon with Thunder Wave, it will pick the latter because electric is super effective on water.

Example 3: The player has a normal type out, the AI just had an electric pokemon faint, and it has a pokemon with Hyper Beam and a pokemon with Thunderbolt, it will pick the latter because the pokemon it just had faint has STAB with Thunderbolt.

Example 4: The player has a Scizor out, and the initial damage roll after stats and STAB was 31. There is an electric move, a fighting move, and a rock move that are checked. For the electric move it is neutral and left unchanged at 31 damage, for the rock move it first doubles for bug and then halves for steel, giving 31 damage. However, for fighting it first halves for bug giving

15, then doubles with steel giving 30. As a result it will see the fighting move as lower damage than the other two choices, despite all of these being neutral damage.

Non-Standard Damage Moves:

OHKO Moves (Fissure, Horn Drill, Guillotine, Sheer Cold)

Variable Power Moves (Flail, Frustration, Low Kick, Magnitude, Present, Return, Reversal)

Reflecting Moves (Counter, Mirror Coat)

Flat Damage Moves (Dragon Rage, Endeavor, Night Shade, Psywave, Seismic Toss, Sonic Boom, Super Fang)

Bide and Hidden Power

Note: In Emerald Kaizo, Hidden Power is implemented as a series of moves so should be counted as a normal damaging move.

Type interactions order:

Attacking type, defending type, effectiveness multiplier

NORMAL ROCK 0.5

NORMAL STEEL 0.5

FIRE FIRE 0.5

FIRE WATER 0.5

FIRE GRASS 2.0

FIRE ICE 2.0

FIRE BUG 2.0

FIRE ROCK 0.5

FIRE DRAGON 0.5

FIRE STEEL 2.0

WATER FIRE 2.0

WATER WATER 0.5

WATER GRASS 0.5

WATER GROUND 2.0

WATER ROCK 2.0

WATER DRAGON 0.5

ELECTRIC WATER 2.0

ELECTRIC ELECTRIC 0.5

ELECTRIC GRASS 0.5

ELECTRIC GROUND 0.0

ELECTRIC FLYING 2.0

ELECTRIC DRAGON 0.5

GRASS FIRE 0.5

GRASS WATER 2.0

GRASS	GRASS	0.5
GRASS	POISON	0.5
GRASS	GROUND	2.0
GRASS	FLYING	0.5
GRASS	BUG	0.5
GRASS	ROCK	2.0
GRASS	DRAGON	0.5
GRASS	STEEL	0.5
ICE	WATER	0.5
ICE	GRASS	2.0
ICE	ICE	0.5
ICE	GROUND	2.0
ICE	FLYING	2.0
ICE	DRAGON	2.0
ICE	STEEL	0.5
ICE	FIRE	0.5
FIGHTING	NORMAL	2.0
FIGHTING	ICE	2.0
FIGHTING	POISON	0.5
FIGHTING	FLYING	0.5
FIGHTING	PSYCHIC	0.5
FIGHTING	BUG	0.5
FIGHTING	ROCK	2.0
FIGHTING	DARK	2.0
FIGHTING	STEEL	2.0
POISON	GRASS	2.0
POISON	POISON	0.5
POISON	GROUND	0.5
POISON	ROCK	0.5
POISON	GHOST	0.5
POISON	STEEL	0.0
GROUND	FIRE	2.0
GROUND	ELECTRIC	2.0
GROUND	GRASS	0.5
GROUND	POISON	2.0
GROUND	FLYING	0.0
GROUND	BUG	0.5
GROUND	ROCK	2.0
GROUND	STEEL	2.0
FLYING	ELECTRIC	0.5
FLYING	GRASS	2.0
FLYING	FIGHTING	2.0
FLYING	BUG	2.0
FLYING	ROCK	0.5

FLYING	STEEL	0.5	
PSYCHIC	FIGHTING		2.0
PSYCHIC	POISON		2.0
PSYCHIC	PSYCHIC		0.5
PSYCHIC	DARK	0.0	
PSYCHIC	STEEL	0.5	
BUG	FIRE	0.5	
BUG	GRASS		2.0
BUG	FIGHTING		0.5
BUG	POISON		0.5
BUG	FLYING		0.5
BUG	PSYCHIC		2.0
BUG	GHOST		0.5
BUG	DARK	2.0	
BUG	STEEL	0.5	
ROCK	FIRE	2.0	
ROCK	ICE	2.0	
ROCK	FIGHTING		0.5
ROCK	GROUND		0.5
ROCK	FLYING		2.0
ROCK	BUG	2.0	
ROCK	STEEL	0.5	
GHOST	NORMAL		0.0
GHOST	PSYCHIC		2.0
GHOST	DARK	0.5	
GHOST	STEEL	0.5	
GHOST	GHOST		2.0
DRAGON	DRAGON		2.0
DRAGON	STEEL	0.5	
DARK	FIGHTING		0.5
DARK	PSYCHIC		2.0
DARK	GHOST		2.0
DARK	DARK	0.5	
DARK	STEEL	0.5	
STEEL	FIRE	0.5	
STEEL	WATER		0.5
STEEL	ELECTRIC		0.5
STEEL	ICE	2.0	
STEEL	ROCK	2.0	
STEEL	STEEL	0.5	
NORMAL	GHOST		0.0
FIGHTING	GHOST		0.0