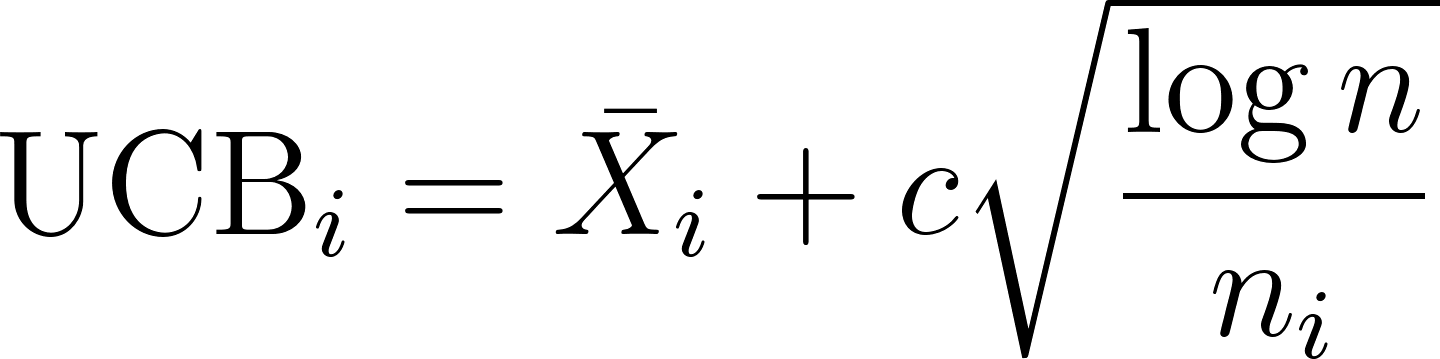
**USB-EpsilonGreedy solution:**

This new combined method is an adaptive selection hyperheuristic combining the Upper Confidence Bound (UCB) and Epsilon-Greedy algorithms from Reinforcement Learning, designed to solve the BPP by continuously learning and adjusting low-level heuristics based on their performance feedback.

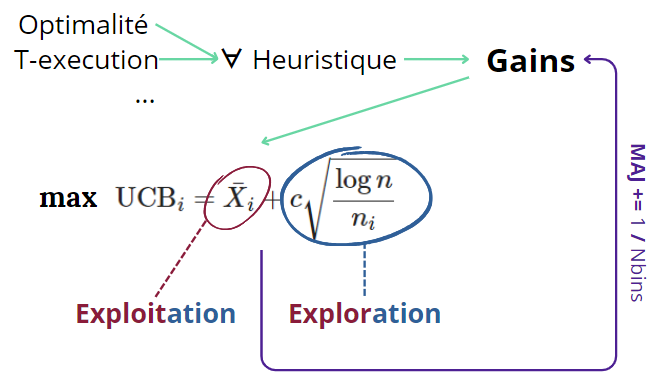
The UCB method balances exploration and exploitation by calculating an upper confidence bound for each heuristic, guiding the selection process towards those with higher potential rewards while ensuring that less frequently used heuristics are also explored. Specifically, the UCB formula used is:

[](https://www.codecogs.com/eqnedit.php?latex=%20%5Ctext%7BUCB%7D_i%20%3D%20%5Cbar%7BX%7D_i%20%2B%20c%20%5Csqrt%7B%5Cfrac%7B%5Clog%20n%7D%7Bn_i%7D%7D%20#0)

where [](https://www.codecogs.com/eqnedit.php?latex=%5Cbar%7BX%7D_i#0) represents the average reward of heuristic [](https://www.codecogs.com/eqnedit.php?latex=i#0), [](https://www.codecogs.com/eqnedit.php?latex=c#0) is a constant typically [](https://www.codecogs.com/eqnedit.php?latex=%5Csqrt%7B2%7D#0) that adjusts the balance between exploration and exploitation, [](https://www.codecogs.com/eqnedit.php?latex=n#0) is the total number of selections, and [](https://www.codecogs.com/eqnedit.php?latex=n_i#0) is the number of times the heuristic [](https://www.codecogs.com/eqnedit.php?latex=i#0) has been selected.

* **Exploration**: UCB encourages trying less-used heuristics by assigning a higher exploration bonus, called [](https://www.codecogs.com/eqnedit.php?latex=%5CDelta_i%20%3D%20%5Ctext%7BUCB%7D_i%20-%20%5Cbar%7BX%7D_i%20#0) This helps discover potentially better placement strategies and escape local optima.
* **Exploitation**: UCB favors selecting high-performing heuristics based on past success (measured by a reward function) [](https://www.codecogs.com/eqnedit.php?latex=%5Cbar%7BX%7D_i#0).

To enhance diversification, we integrate the Epsilon-Greedy strategy. With a predefined probability [](https://www.codecogs.com/eqnedit.php?latex=%5Cvarepsilon#0), the algorithm randomly selects a heuristic to promote exploration. Otherwise, it chooses the heuristic with the highest UCB value, ensuring effective exploitation of the best-known heuristics.



**TS Hyperheuristic:**

In this approach, the high-level strategy is implemented as a Tabu Search (TS) algorithm. The process begins by generating an initial combination of low-level heuristics randomly. During each iteration, the algorithm generates neighboring solutions by modifying the current combination of heuristics by swapping them. The neighbor that most improves the solution is selected as the new current solution. To prevent cycling back to previous solutions, the current solution is added to a tabu list. The combination of low-level heuristics is then executed sequentially, with each heuristic addressing a part of the problem.

The Tabu Search mechanism provides a good balance between exploration (searching new areas of the solution space) and exploitation (refining known good solutions). The tabu list helps in avoiding local optima and cycling back to previous solutions. The tabu list effectively prevents the search from revisiting previously explored solutions, ensuring that the algorithm continually progresses towards new and potentially better solutions.