Opis działania algorytmu:

Na początku generowane zostaje rozwiązanie początkowe (pierwszy potencjalny kandydat), przy pomocy algorytmu zachłannego deterministycznego lub algorytmu zachłannego z pierwszym wierzchołkiem losowym i dodane do kolejki Tabu.

Następnie w pętli trwającej założoną ilość czasu algorytm szuka optymalnego rozwiązania.

Na podstawie obecnego „potencjalnego kandydata” (rozwiązania, które jest najlepsze spośród sąsiedztwa) generowane jest sąsiedztwo rozwiązania, którego wielkość można regulować. Wśród sąsiedztwa są rozwiązania będące zmodyfikowanym obecnym kandydatem. Modyfikuje się je poprzez wykonywanie tzw. „tabu moves”. Nad każdym rozwiązaniem z sąsiedztwa wykonuje się pewną, ustaloną wcześniej liczbę tabu moves.

Tabu move użyty w algorytmie działa na zasadzie „Vertex Insertion”, tzn. że wybiera losowo wierzchołek z losowo wybranej trasy, a następnie szuka potencjalnej trasy (różnej od losowo wybranej), gdzie może umieścić wybrany wierzchołek do trasy, najczęściej w momencie, kiedy przybycie (uwzględniając czekanie do początku okna czasowego) do tego wybranego wierzchołka wymaga mniej czasu niż przybycie do wierzchołka następnego w znalezionej trasie. Jeśli rozwiązanie jest prawidłowe, trasy zostają odpowiednio zmodyfikowane i tabu move kończy działanie. Jeśli trasa, do której można wstawić losowy wierzchołek nie zostanie znaleziona, wtedy rozwiązanie nie zostanie zmienione.

Po wygenerowaniu sąsiedztwa, każde zawarte w nim rozwiązanie zostaje poddane analizie, czy jest ono lepsze od obecnego kandydata i czy nie należy do kolejki Tabu. Jeśli tak, to te rozwiązanie zostaje nowym kandydatem. Jeśli rozwiązanie jest lepsze od wcześniej znalezionego, to zostaje także zapisane do pamięci. Warunek sprawdzenia, czy rozwiązanie należy do kolejki Tabu ma na celu uniknięcie tworzenia tych samych, poprzednich rozwiązań i krążenia wokół lokalnego optimum. Własność ta nazywana jest „short-term memory”.

Pojemność liczby Tabu jest ograniczona i możliwa do wyregulowania, aby móc decydować, jak algorytm „daleko” ma szukać potencjalnych nowych rozwiązań od znalezionego lokalnego optimum.

Po zakończeniu głównej pętli algorytmu, rozwiązanie, które algorytm zwraca jest najlepszym znalezionym rozwiązaniem przez cały okres działania tej pętli.

Parametry do regulacji:

* Czas działania głównej pętli
* Pojemność kolejki Tabu
* Wielkość przeszukiwanego sąsiedztwa
* Liczba „tabu moves” użytych do znalezienia sąsiedniego rozwiązania