

Introdução:

- Diante do desafio apresentado sobre o Problema de Coleta e Entrega com Objetivos Múltiplos e Restrições de Tempo (MOCPDPTW) apresentado construímos a seguinte abordagem em forma de um pseudocódigo:
- 1 - Inicialize uma lista "rotas" com K elementos vazios.
 - 2 - Defina "solicitações" como um conjunto de pares de coleta e entrega.
 - 3 - Para cada rota em "rotas", inicialize-a com uma solicitação aleatória escolhida de "solicitações".
 - 4 - Remova cada solicitação inserida de "solicitações".
 - 5 - Enquanto não houver todas as solicitações inseridas: Inicialize "novaSolução" como nula.
 - 6 - Escolha uma solicitação usando H.
 - 8 - Para cada índice idx e rota na lista "rotas":
 9. Insira a solicitação escolhida na rota atual usando I.
 10. Melhore a rota usando O.
 11. Se a rota é viável:
 12. Se "novaSolução" é nula:
 13. Atribua "rotas" a "novaSolução".
 14. Atribua "novaSolução[idx]" a rota atual.
 - 9 - Se "novaSolução" é melhor que "rotas":
 16. Atribua "novaSolução" a "rotas".
 - 10 - Se "novaSolução" é nula:
 18. Escolha uma solicitação aleatória de "solicitações".
 19. Adicione uma nova rota à lista "rotas" contendo a solicitação escolhida.
 - 11 - Remova a solicitação escolhida de "solicitações".
 - 12 - Retorne "rotas".

Explicação:

Este é o pseudocódigo de um algoritmo para solucionar um problema de roteamento de entregas, usando uma heurística de inserção (H), um operador de inserção (I), um operador de busca local (O) e um número especificado de rotas (K). O algoritmo inicializa uma lista "rotas" com K elementos vazios e define "solicitações" como um conjunto de pares de coleta e entrega. Em seguida, para cada rota na lista "rotas", uma solicitação aleatória é escolhida de "solicitações" e inserida na rota. Cada solicitação inserida é então removida de "solicitações". O algoritmo então entra em um loop, que continua enquanto houver solicitações restantes para serem inseridas. Dentro do loop, uma solicitação é escolhida usando a heurística H. Em seguida,

para cada índice e rota na lista "rotas", a solicitação é inserida na rota atual usando o operador I, e a rota é melhorada usando o operador O. Se a rota resultante é viável, ela é comparada com a atual solução, sendo assim atribuída, se for melhor. Se não houver solução viável, uma nova rota é adicionada à lista "rotas" contendo uma solicitação aleatória escolhida de "solicitações". O loop continua até que todas as solicitações tenham sido inseridas, e então a lista "rotas" é retornada como solução viável.