

# Projecto Inteligência Artificial – Problema de Optimização de Espaço

<https://github.com/DAE-2020/IA-StockingProblem>

Joel Rodrigues  
Estudante em

Escola Superior de Tecnologia e Gestão de Leiria  
2191999@my.pleiria.pt

Marc Bernardo  
Estudante em

Escola Superior de Tecnologia e Gestão de Leiria  
2192824@my.ipleiria.pt

## 1. INTRODUÇÃO

Este projeto está relacionado com Problema de Armazenamento, o que se prevê que sejam encontradas as melhores soluções com base em algoritmos genéticos, recombinações e mutações. Para o mesmo foram dados os seguintes exemplos práticos de peças com diferentes tamanhos e formas, representadas binariamente, que iriam dar lugar a uma fila com N colunas e X linhas, onde o X é dado através de DataSets. Foi decidido a utilização destas recombinações de forma a possibilitar, com base em probabilidades, a criação de sucessores mais aptos.

## 2. OPERADORES DE RECOMBINAÇÃO

### 2.1 Complete Subtour Exchange Crossover (CSEX)

No âmbito dos operadores de recombinação, optamos por comparar o desempenho do Complete Subtour Exchange Crossover (CSEX), sendo este um operador apropriado a problemas do tipo “Caxeiro Viajante”, já que são usadas as melhores rotas (com melhor fitness) para serem propagadas para os indivíduos sucessores.

O algoritmo do CSEX procura *subtours* comuns entre os progenitores, pois estes apresentam-se geralmente como genes que resultam em indivíduos ideais. Depois de encontrados todos os *subtours* (condição necessária neste operador de recombinação), são gerados todos os sucessores resultantes de todas as combinações possíveis de *subtours*. Por fim e para manter fixo o número de indivíduos por geração, é usada a função de avaliação para selecionar os dois melhores sucessores que irão representar os seus antecedentes.

Na demonstração teórica do algoritmo, considerando os progenitores [1, 2, 4, 3, 5, 6] e [2, 1, 3, 5, 6, 4]. Na recombinação destes indivíduos, podemos identificar como *subtours*, os seguintes conjuntos, [1, 2] e [3, 5, 6] no primeiro progenitor e consequentemente [2, 1] e [3, 5, 6] no segundo progenitor.

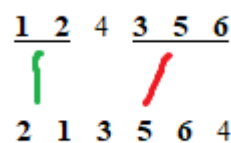
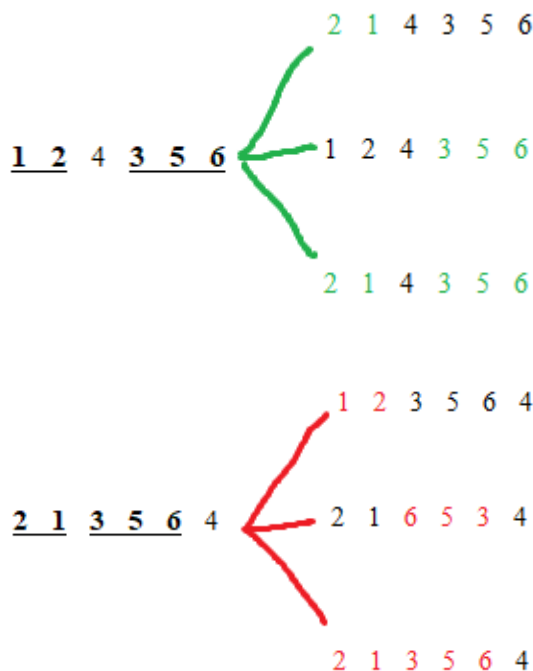


Figura 1 Procura de *subtours*

De seguida, são geradas todas as combinações de *subtours* de ambos os progenitores. Neste processo são gerados  $2^{(n+1)} - 2$ , sendo N o número de *subtours* encontrados. Neste caso, foram encontrados dois *subtours* que resultam nos seguintes sucessores, [2, 1, 4, 3, 5, 6], [1, 2, 4, 6, 5, 3] e [2, 1, 4, 6, 5, 3] do primeiro progenitor, e [1, 2, 3, 5, 6, 4], [2, 1, 6, 5, 3, 4] e [1, 2, 6, 5, 3, 4] do segundo progenitor.



Finalmente para que a geração preserve o número de indivíduos, são escolhidos através da função de avaliação, os 2 sucessores mais aptos.

## 2.2 Order based Crossover (OX2)

Para este projeto pretendemos demonstrar alternativas para a recombinação PMX com isto uma das mais parecidas seria a OX2. Para este caso iremos mostrar um exemplo prático, com 2 indivíduos:

[10, 4, 7, 9, 1, 5, 8, 2, 3, 6], [4, 8, 6, 3, 1, 2, 9, 10, 5, 7]

foram escolhidas 3 posições aleatórias, ordenadas, posição 0, 3, 7. Com estas posições são guardados os seus valores: 10, 9, 2 e 4, 3, 10 respetivamente. Para processar o crossover, são guardados estes valores e são trocados nas respetivas posições tirando as exceções que constam nas imagens, em que iremos representar passo a passo.

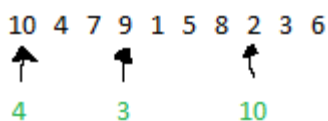


Figura 1. Sem substituição.

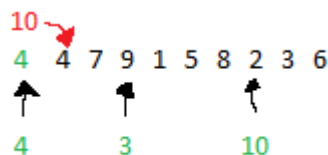


Figura 2. Primeira substituição e não corrigido.

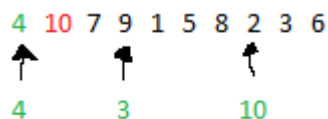


Figura 3. Primeira substituição e corrigido.

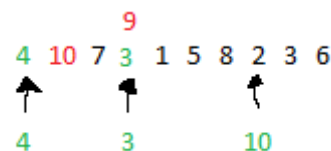


Figura 4. Segunda substituição e não corrigido.

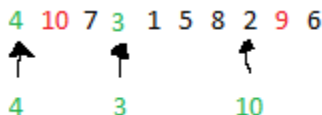


Figura 5. Segunda substituição e corrigido.

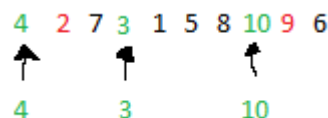


Figura 6. Terceira substituição, sobreposição e correção.

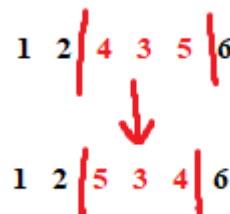
O mesmo se segue para o segundo pai resultando em:

[10, 8, 6, 9, 1, 4, 3, 2, 5, 7]

## 3. OPERADORES DE MUTAÇÃO

### 3.1 Inversion

À semelhança da mutação genética conhecida pela biologia, no processo de mutação por inversão, um excerto do genoma (admitindo o caso em que cada indivíduo possui apenas um cromossoma) é invertido e posteriormente ligado ao restante código genético.



### 3.2 Swap

Esta mutação afeta apenas duas posições do genoma de um indivíduo aleatórias, mantendo o restante intato. A única operação que é feita é troca do valor de uma posição pela outra.

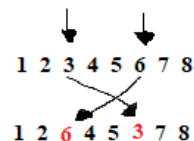


Figura 7. Operação de troca

## 4. Bibliografia

- A.J. Umbarkar, P. S. (2015). *Crossover Operators in Genetic Algorithms: A Review*. ICTACT.
- KENGO KATAYAMA, H. H. (2003). *Analysis of Crossovers and Selections in a Coarse-Grained Parallel Genetic Algorithm*. Elsevier.