Questão 1

Capítulo 9, Exercício 4:

Describe the main components necessary to add to a "standard" EA in order to tackle a multiobjective problem.

O problema multiobjetivo pode ser pensado como um problema que possui uma função custo global, a qual é formada através da combinação de outras funções custo. Idealmente, minimizar a função global consistiria em minimizar, individualmente, cada função que a compõe, porém, na prática, as funções geradoras competem entre si, ou seja, otimizar uma delas implica em degenerar o resultado de outra. Dessa forma, o que o algoritmo evolucionário busca é um equilíbrio entre elas, o qual gera soluções "ótimas", segundo um conjunto de fatores.

Visando esse objetivo, a primeira modificação a se realizar no algoritmo de EA tradicional seria introduzir um mecanismo de preservação de diversidade da população. Essa condição surge naturalmente da essência dos problemas multiobjetivos, que é a existência de um *trade-off* entre as diferentes funções que se busca minimizar. Tal preservação pode ser feita de forma implicíta, por exemplo, com os conceitos de *Speciation ("especiação")* (somente indivíduos de uma mesma espécie podem reproduzir e competir entre si, permitindo a coexistência de diversas espécies), ou de *Punctuated Equilibria* (indivíduos de uma mesma espécie, mas pertencentes a diferentes subpopulações, migram para outra subpopulação e se recombinam com indivíduos dela, garantindo a mistura de diferentes soluções encontradas em cada subpopulação); ou de forma explicíta, com os conceitos de *Crowding* (filhos e pais mais similares competem entre si, sobrevivendo o melhor dos dois, garantindo um número igual de indivíduos em cada nicho de solução), ou *Fitness-Sharing* (indivíduos de uma mesma região compartilham um mesmo valor de aptidão, sobre o qual a seleção de sobreviventes é feita, permitindo a preservação de indivíduos pertencentes a diferentes nichos).

Além disso, uma segunda medida a se tomar seria considerar a dominância das soluções da população sobre as outras, moldando os critérios de seleção de sobreviventes sobre tal, em vez de considerar valores absolutos das funções custo. O conceito de dominância é definido da seguinte forma: Supondo duas funções custo globais, A e B, as quais, por sua vez, são compostas por n funções custo; então, diz-se que A domina B, se, em todas as n funções custo, o valor de cada uma delas, em A, é, no mínimo, igual ou maior ao respectivo valor de cada uma delas em B e o valor de pelo menos uma delas, em A, é maior do que o respectivo valor em B. Esse cuidado visa buscar a chamada *Fronte de Pareto*, que é um conjunto formado por todas as soluções não-dominadas do problema. Essas soluções são pensadas como soluções "ótimas", levando-se em conta o trade-off entre os diferentes objetivos que se deseja minimizar. Desse conjunto, cabe ao usuário selecionar a melhor, segundo suas preferências em relação a cada objetivo. Conforme dito anteriormente, basear o critério de seleção de sobreviventes sobre a dominância de soluções permite uma busca mais efetiva do Fronte de Pareto. Algumas abordagens para localizar o Fronte de Pareto incluem:

- Associar um valor comum de fitness a indivíduos, baseado na quantidade de soluções que eles dominam mais 1;

- Encontrar os frontes mediante uma busca, na população atual, de todas as soluções não dominadas, que ainda não foram incluídas em nenhum fronte e, então, associar um valor de fitness a soluções desse fronte, baseado na quantidade de soluções pertencentes a frontes inferiores (ou seja, dominados) pela solução atual;

- Realizar um torneio para a seleção de sobreviventes, baseado, primeiro, na dominância de uma solução sobre a outra e, depois, na quantidade de soluções similares que se encontram na população;

- Realizar um torneio para a seleção de pais, nos mesmos moldes da seleção de sobreviventes do item anterior.

Em resumo, os principais componentes a se adicionar no EA clássico, de modo a resolver problemas multiobjetivo, seriam: um controle (implícito ou explícito) de diversidade da população e alterar os esquemas de seleção de sobreviventes (e, possivelmente, de pais), de modo a utilizarem a dominância entre soluções como critério de seleção, em vez de valores absolutos das funções fitness (como é comumente utilizado no EA clássico, para problemas de um só objetivo).

Questão 2

Capítulo 10, Exercício 2 (só a primeira frase do enunciado):

Implement a simple memetic algorithm using a single iteration of a bit-flipping local search within the code for the SGA you developed for One-Max in Chapter 3. *Before* you run the experiment, justify whether you think steepest or greedy ascent will be most efficient for

this problem.

Questão 3

Capítulo 12, Exercício 1:

Specify the eight-queens problem as a CSP $\langle S, \phi \rangle$.

O problema das oito rainhas é naturalmente pensado como um problema de satisfação de restrições, já que somente são consideradas soluções válidas aquelas, nas quais nenhuma das oito rainhas é atacada por outra (ou seja, não outras rainhas presentes nas direções vertical, horizontal e diagonal da rainha em questão). De modo a formalizar a definição desse problema como um CSP (*"Constrained Satisfaction Problem"*), define-se dois parâmetros: o espaço de busca S e a restrição ϕ a ser respeitada.

1. Definição do espaço de busca S

O espaço de busca S, ao qual pertencem as possíveis soluções do problema, pode ser definido de diversas formas. Uma maneira, por exemplo, seria $S = (l, c)^8$, onde $l, c \in \{1, 2, 3, 4, 5, 6, 7, 8\}$. Nesse caso, como o tabuleiro de xadrez tem tamanho 8×8, são precisos 8 pares-ordenados (l, c) para definir uma solução, já que cada um deles define a posição de uma rainha (linha e coluna, respectivamente). Vale ressaltar que esse espaço de soluções apresenta uma quantidade elevada de soluções inválidas, uma vez que nenhuma restrição é implicitamente imposta sobre ele. Caso se adote esse espaço, o controle sobre as restrições do problema deverá ser feito de forma explícita, ou seja, não aceitar soluções desse espaço que infrinjam alguma restrição. Para exemplificar, a solução $\{(1, 2), (1, 3), (2, 2), (5, 5), (8, 4), (7, 1), (4, 6), (3, 8)\}$ não seria válida, já que mais de uma rainha em uma linha ou coluna não é permitido.

Alternativamente, é possível adicionar implicitamente uma restrição no espaço de soluções, de modo que o novo espaço gerado apresenta, somente, soluções que automaticamente respeitam a restrição imposta. Sendo assim, o espaço de busca poderia ser definido como $S = C^8$, onde $C = \{1, 2, 3, 4, 5, 6, 7, 8\}$. Nesse caso, C define a coluna sobre a qual uma rainha será colocada (como são 8 colunas, o espaço de busca fica C^8), ao passo que os números que definem esse conjunto correspondem ao índice da linha sobre a qual a rainha da coluna se situa. Note que, com essa nova definição de espaço, a restrição de que nenhuma coluna pode conter mais de uma rainha é automaticamente satisfeita.

A definição anterior contempla a restrição das colunas, porém, deixa em aberto a restrição das linhas e das diagonais, sendo necessário, portanto, a validação explicíta dessas condições, antes do algoritmo aceitar tais soluções. No entanto, é possível, ainda, definir o espaço de uma outra forma, de modo a incluir a restrição das linhas (nenhuma linha pode conter mais do que uma rainha), através da seguinte forma: $S = \{s_1, s_2, \ ..., s_8\}$, onde $s_i \in \{1, 2, 3, 4, 5, 6, 7, 8\}$ e $\forall i, j \in \{1, 2, 3, 4, 5, 6, 7, 8\}, s_i \neq s_j$. O índice i representa o número da coluna, enquanto que s_i representa o número da linha em que se localiza a rainha da coluna i. Em outras palavras, o espaço anterior é definido como uma permutação de números não-repetidos entre 1 e 8. Nesse caso, a restrição das linhas é automaticamente satisfeita, pois, como os números não são repetidos, não há como soluções desse espaço representarem rainhas em linhas iguais. Adotando esse espaço reduzido como o de busca, resta somente ao algoritmo checar explicitamente a condição das diagonais, antes de validar uma solução. A vantagem de definir restrições implicitamente é que isso poupa tempo de processamento do algoritmo, uma vez que ele gasta menos tempo na região de soluções inválidas do problema.

2. Definição da restrição ϕ

A restrição ϕ para o problema das oito rainhas pode ser definida como uma composição de 3 restrições, conforme já apontado anteriormente: de linhas, de colunas e de diagonais.

1. $r_l = verdadeiro$, se não houver mais do que uma rainha presente em uma linha;
2. $r_c = verdadeiro$, se não houver mais do que uma rainha presente em uma coluna;
3. $r_d = verdadeiro$, se não houver mais do que uma rainha presente em uma diagonal.

Dessa forma, a restrição ϕ será verdadeira, se, e somente se, as 3 outras restrições forem verdadeiras: $\phi(\overline{s}) = verdadeiro \iff r_l \wedge r_c \wedge r_d = verdadeiro$ ($\wedge$ é o operador lógico matemático *"AND"* ("E")).

Com a caracterização dos parâmetros S e ϕ, conclui-se a especificação do problema das oito rainhas como um CSP.

Questão 4

Escreva (utilizando um pseudo-código) um algoritmo genético simples para a solução do Exercício 4 da Lista de Exercícios #5, considerando que a população não é de dicionários $\mathbf{Y}$, mas sim de matrizes $p_{\mathbf{y}|\mathbf{x}}$ descrevendo as probabilidades com as quais um vetor de dados $\mathbf{x}$ é atribuído a um agrupamento $\mathbf{y}$.