

Universidade de Brasília

Instituto de Física

Davi Valadares Rodrigues Feliciano

Trabalho de Conclusão de Curso

**Algoritmo Genético para o Ajuste das Bandas de
Energia do CrS₂ e CrSe₂ via Modelo $k \cdot p$**

Brasília

22 de setembro de 2022

Davi Valadares Rodrigues Feliciano

**Algorítmo Genético para o Ajuste das Bandas de Energia
do CrS₂ e CrSe₂ via Modelo $k \cdot p$**

Trabalho de Conclusão de Curso apresentado para obtenção do grau de Bacharel em Física pelo Instituto de Física da Universidade de Brasília.

Universidade de Brasília
Instituto de Física

Orientador: Luiz Antonio Ribeiro Junior

Brasília
22 de setembro de 2022

Resumo

Nas diversas áreas do conhecimento humano é bem conhecida uma série de problemas de otimização. A solução desses problemas consiste em encontrar em um conjunto de configurações, um subconjunto, ou mesmo um elemento que melhor satisfaça um ou mais vínculos previamente determinados. Uma estratégia famosa por solucionar problemas de tal classe de forma rápida e eficiente consiste no emprego dos algoritmos genéticos. São assim chamados devido à forte inspiração em fenômenos da biologia evolutiva — como mutação, recombinação (ou *crossover*) e seleção — na elaboração de suas etapas de execução. Neste projeto de trabalho é proposto um algoritmo genético para a otimização de funções reais, isto é, para a procura dos pontos nos quais a função é máxima (ou mínima). Sua implementação é feita em Python com o uso da biblioteca NumPy. O algoritmo foi aplicado na otimização de algumas funções de exemplo e na busca de uma matriz hamiltoniana do modelo $k \cdot p$ para o melhor ajuste das bandas de energia de condução e de valência de alguns Dicalcogenetos de Metais de Transição — CrS_2 e CrSe_2 — calculadas previamente via o método DFT. Por fim, é feita uma discussão acerca dos resultados e da eficácia e performance do programa.

Palavras-chaves: Algoritmo Genético, Otimização Numérica, Máximos de Funções, Mínimos de Funções, Ajuste de Curvas, Dicalcogenetos de Metais de Transição, Python.

Abstract

On the diverse areas of human knowledge, there are a series of well-known optimization problems. The solution of these problems consists in finding, in a collection of configurations, a subset, or even an element that better satisfies one or more constraints previously determined. A famous strategy to solve such problems in a quick and efficient fashion relies on the use of genetic algorithms. They receive such name due to the strong inspiration in phenomena of evolutionary biology—like mutation, recombination (or crossover) and selection—in its execution. In this project, a genetic algorithm is proposed for the optimization of real functions, that is, for the search of the points in which the function is maximum (or minimum). Its implementation is done using the Python programming language, by means of the NumPy library. The algorithm developed was applied in the optimization of some example functions and in the search for an hamiltonian matrix of the $k \cdot p$ model that best fits the valence and conduction bands of some transition metal dichalcogenides— CrS_2 e CrSe_2 —calculated beforehand by means of the DFT technique. Finally, a discussion was conducted concerning the results, as well as on the efficiency and performance of the algorithm implementation.

Keywords: Genetic Algorithm, Numerical Optimization, Maximum of Functions, Minimum of Functions, Curve Fitting, Transition Metal Dichalcogenides, Python.

Listas de ilustrações

Figura 1 – Gráfico da função definida na Equação 1.1, com $n = 9$ e $\sigma = 0,4$.	9
Figura 2 – Fluxograma geral de um algoritmo genético.	11
Figura 3 – (a) Molibdenita incrustada em uma massa de quartzo. (b) Representação da estrutura cristalina da molibdenita. Molibdênio em azul, enxofre em amarelo.	12
Figura 4 – Zona de Brillouin para monocamadas de TMDCs e principais pontos de simetria. Geralmente as bandas de valência e de condução são calculadas ao longo desses pontos, como no caminho exemplificado.	12
Figura 5 – Energias das bandas de condução e de valência ao longo do caminho ilustrado na Figura 4 para o dissulfeto de molibdênio.	14
Figura 6 – Diagrama ilustrando —em um caso com $m = 1$ e $l = 8$ — a recombinação entre os indivíduos S_k e S_{k+1} e mutação nos filhos gerados segundo as respectivas máscaras de recombinação e mutação R e M .	21
Figura 7 – (a) Gráfico da função $f_1(x, y)$. (b) Gráfico da função $f_2(x, y)$.	25
Figura 8 – Curvas de nível da função $f_1(x, y)$. Os pontos em preto indicam as posições dos indivíduos de 8 populações em sua 100 ^a geração na otimização da função, com $p_2 = p_3 = 5\%$. Marcado com um × vermelho estão os melhores indivíduos de cada população.	26
Figura 9 – Evolução dos valores da função $f_1(x, y)$ para os melhores 200 indivíduos de cada população, diferenciadas por cor, em termos da geração g , com $p_2 = p_3 = 5\%$.	27
Figura 10 – Curvas de nível da função $f_1(x, y)$. Os pontos em preto indicam as posições dos indivíduos de 8 populações em sua 100 ^a geração na otimização da função, com $p_2 = p_3 = 20\%$. Marcado com um × vermelho estão os melhores indivíduos de cada população.	28
Figura 11 – Evolução dos valores da função $f_1(x, y)$ para os melhores 200 indivíduos de cada população, diferenciadas por cor, em termos da geração g , com $p_2 = p_3 = 20\%$.	29
Figura 12 – Curvas de nível da função $f_2(x, y)$. Os pontos em preto indicam as posições dos indivíduos de 8 populações em sua 100 ^a geração na otimização da função, com $p_2 = p_3 = 5\%$. Marcado com um × vermelho estão os melhores indivíduos de cada população.	30
Figura 13 – Evolução dos valores da função $f_2(x, y)$ para os melhores 200 indivíduos de cada população, diferenciadas por cor, em termos da geração g , com $p_2 = p_3 = 5\%$.	31

Figura 14 – Curvas de nível da função $f_2(x, y)$. Os pontos em preto indicam as posições dos indivíduos de 8 populações em sua 100 ^a geração na otimização da função, com $p_2 = p_3 = 20\%$. Marcado com um × vermelho estão os melhores indivíduos de cada população.	32
Figura 15 – Evolução dos valores da função $f_2(x, y)$ para os melhores 200 indivíduos de cada população, diferenciadas por cor, em termos da geração g , com $p_2 = p_3 = 20\%$	33
Figura 16 – Gráficos das bandas de energia ajustadas para CrS ₂ via Algoritmo Genético (a) e via <i>Dual Annealing</i> (b) usando as expansões de 1 ^a e 3 ^a ordem de \hat{H}_{kp}	37
Figura 17 – Gráficos das bandas de energia ajustadas para CrSe ₂ via Algoritmo Genético (a) e via <i>Dual Annealing</i> (b) usando as expansões de 1 ^a e 3 ^a ordem de \hat{H}_{kp}	38
Figura 18 – Gráficos dos níveis de Landau no vale K para os materiais (a) CrS ₂ e (b) CrSe ₂ em termos da magnitude do campo magnético externo aplicado na direção normal a respectiva monocamada.	39
Figura 19 – Gráficos dos níveis de Landau no vale K' para os materiais (a) CrS ₂ e (b) CrSe ₂ em termos da magnitude do campo magnético externo aplicado na direção normal a respectiva monocamada.	40

Listas de tabelas

Tabela 1 – Intervalos de busca para cada parâmetro considerados no ajuste das bandas de energia via modelo $k \cdot p$. Definidos estes intervalos, a região de busca é calculada conforme a Equação 2.5.	34
Tabela 2 – Valores para os parâmetros de rede considerados nos processos de otimização e valores esperados para o <i>bandgap</i> , para cada um dos materiais considerados.	34
Tabela 3 – Parâmetros da hamiltoniana \hat{H}_{kp} ajustados para CrS_2 (a) e para CrSe_2 (b) usando as expansões de 1 ^a e 3 ^a ordem de \hat{H}_{kp} , bem como os valores para a função objetivo f correspondente.	36

Sumário

1	INTRODUÇÃO	8
1.1	Problemas de Otimização	8
1.2	O Algoritmo Genético	10
1.3	Um Exemplo Físico: Ajuste das Estruturas de Banda de Dicalcogêne- tos de Metais de Transição	11
1.4	A Implementação do Algoritmo Genético	16
2	METODOLOGIA	18
2.1	Codificação	18
2.2	Seleção	19
2.3	Recombinação e Mutação	20
2.4	Estratégia Elitista	21
3	RESULTADOS	23
3.1	Testes em Funções	23
3.2	Ajuste das Estruturas de Banda dos Cristais CrS₂ e CrSe₂	34
3.3	Testes de Performance	41
4	CONCLUSÃO	42
5	CRONOGRAMA	43
	REFERÊNCIAS	44

1 Introdução

1.1 Problemas de Otimização

Em matemática e ciência da computação, problemas de otimização consistem naqueles em que a solução é um elemento de um conjunto de candidatos que melhor satisfaz uma determinada série de condições. Um exemplo clássico dessa classe é o problema da mochila (*knapsack problem*, em inglês), no qual procura-se, dentre um conjunto de itens com diversos preços e pesos, o subconjunto que maximiza o valor total. Esse valor é calculado somando-se os preços dos itens postos na mochila.

Entretanto a solução deve respeitar um vínculo: o peso total dos objetos escolhidos não deve exceder o peso máximo suportado pela mochila. Ainda que seja possível encontrar a solução exata desse problema usando algoritmos de programação dinâmica, a complexidade de tais algoritmos é $\mathcal{O}(nw)$, onde n é o número de itens e w é a capacidade da bolsa. Assim, o tempo de execução para muitos itens — que é o caso de interesse, usualmente — pode ser mais longo do que o desejado.

Outro exemplo comum de problema de otimização é o de otimização numérica, do qual constituem solução os pontos de máximo ou mínimo global de uma função $f : \mathbb{R}^m \rightarrow \mathbb{R}$. Tomemos como exemplo simples a função

$$f(x, y) = \cos(n\pi r) \exp\left(-\frac{r^2}{\sigma^2}\right), \quad (1.1)$$

definida no domínio $[-1, 1] \times [-1, 1]$, cujo gráfico se encontra na Figura 1. Os parâmetros n e σ são reais, e modificam a frequência de oscilação da função e a rapidez de decaimento da amplitude, respectivamente. Analiticamente é fácil determinar o ponto de máximo global em $r = 0$ usando métodos de cálculo para funções de múltiplas variáveis. Contudo, não é sempre que pode-se dispor de tal facilidade. Nos casos em que a função objetivo é de maior complexidade, com domínio contido em espaços de maior dimensão; em que a função tem evolução temporal; em que a função possui diversas descontinuidades; ou em que a função depende de variáveis aleatórias, o problema se torna impraticável de resolver de forma analítica. Isto posto, como desenvolver um algoritmo para encontrar a solução?

Uma classe de algoritmos mais simples que se propõem a resolver esse problema são os algoritmos do tipo escalada de morro (*hill climbing*, em inglês). Sugerido pelo nome, o funcionamento desses algoritmos consiste, resumidamente, em sortear um ponto inicial no domínio da função, calcular o gradiente da função no ponto, e seguir na direção resultante por uma distância pré-definida, e repetir esses passos até

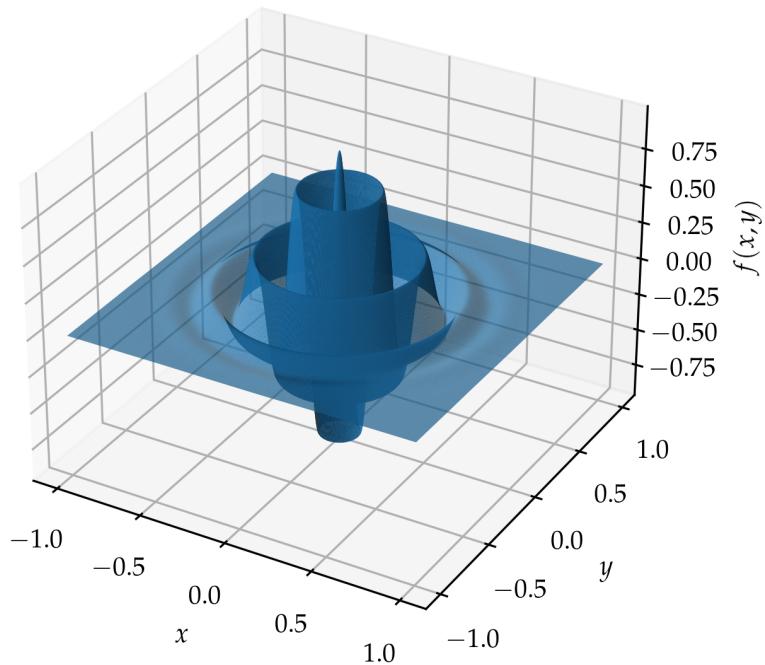


Figura 1 – Gráfico da função definida na Equação 1.1, com $n = 9$ e $\sigma = 0,4$.

que o módulo do vetor gradiente seja próximo de zero, respeitando uma tolerância previamente determinada [1].

O problema com esse tipo de algoritmo é que, quando aplicado em funções como a definida anteriormente, a tendência é de que a solução encontrada seja um máximo local na grande maioria das vezes. No caso da função definida pela Equação 1.1, o máximo global só seria encontrado pelo algoritmo se o ponto inicial ocorresse na região delimitada pelo primeiro mínimo local, isto é, na região* tal que $r < 1/n$. Se os pontos iniciais do algoritmo forem gerados aleatoriamente, para o domínio definido, a probabilidade de um ponto pertencer a esta região é $P(n) = \pi/4n^2$. Para $n = 9$, como na Figura 1, $P \approx 1\%$. Ademais, é possível que esse tipo de algoritmo indique erroneamente como solução pontos de sela e planícies em determinados casos, o que o torna ainda menos eficaz para o propósito.

Uma outra categoria de algoritmos que podem resolver ambos os problemas de forma rápida e chegar a uma solução que se aproxima o suficiente da solução exata são os algoritmos genéticos. Tais algoritmos, quando bem implementados, podem chegar a

*Tal probabilidade é computada tomando a razão entre a área delimitada pelo primeiro mínimo local e a área total do domínio em questão.

uma solução aproximada para o problema da mochila de forma rápida para valores de n e w ordens de grandeza superiores aos que tornariam praticável o uso da estratégia proposta anteriormente.

No problema de otimização numérica, um algoritmo desse tipo é vantajoso pois é capaz de chegar na solução de forma rápida na maioria dos casos, identificando não só o máximo global, como também os máximos locais contidos na região de busca. Outra vantagem é que esse método pode ser aplicado sem problemas em espaços de busca com grande número de dimensões, ou em funções não estacionárias, contínuas ou descontínuas. Além disso, como será mostrado posteriormente, o algoritmo é paralelizável, o que pode acelerar a obtenção de uma solução.

1.2 O Algoritmo Genético

Os algoritmos genéticos foram primeiramente desenvolvidos por John Henry Holland [2] na década de 60. Levam esse nome pois sua formulação teve como forte influência o processo de evolução natural que fomenta a origem das espécies desde o surgimento da vida em nosso planeta. Em sua execução, é inicializada uma população de indivíduos da mesma espécie[†]. Cada indivíduo corresponde a um candidato a solução do problema de otimização proposto, e tem sua identidade codificada pelo seu material genético[‡]. Então, sob algum critério, é selecionada uma parcela desses indivíduos, a qual dará origem a novos descendentes, cujos cromossomos serão gerados pela recombinação dos cromossomos correspondentes nos pais. Nesse passo é introduzida uma probabilidade de mutação nos genes dos filhos.

Ao final desse processo, descarta-se a parcela não selecionada da população, dando origem a uma nova geração, formada pelos pais e seus filhos. Esse processo é repetido iterativamente até que uma condição de parada seja atingida. Essa condição pode ser imposta sobre o número de gerações ou sobre o tempo de execução do programa. Pode ainda ocorrer quando for observado uma determinada característica na população, persistente no decorrer das gerações. No caso de um problema de otimização numérica, a observação de uma porção de indivíduos com seus genes constantes durante um número grande de gerações pode significar a convergência dessa parcela da população para um máximo global, por exemplo. Respeitando as diferenças de implementação em cada passo, devido as peculiaridades de cada tipo

[†]Isso significa que seus respectivos códigos genéticos possuem a mesma estrutura: mesmo número de cromossomos com uma mesma quantidade de genes. Assim, pode haver reprodução entre tais indivíduos.

[‡]No caso do problema da mochila, por exemplo, uma estrutura natural para o material genético é um cromossomo binário, com n genes, cada um correspondente a um objeto do problema. Por outro lado, no problema de otimização numérica, dois cromossomos seriam necessários, um para codificar cada coordenada em cada dimensão do espaço de busca.

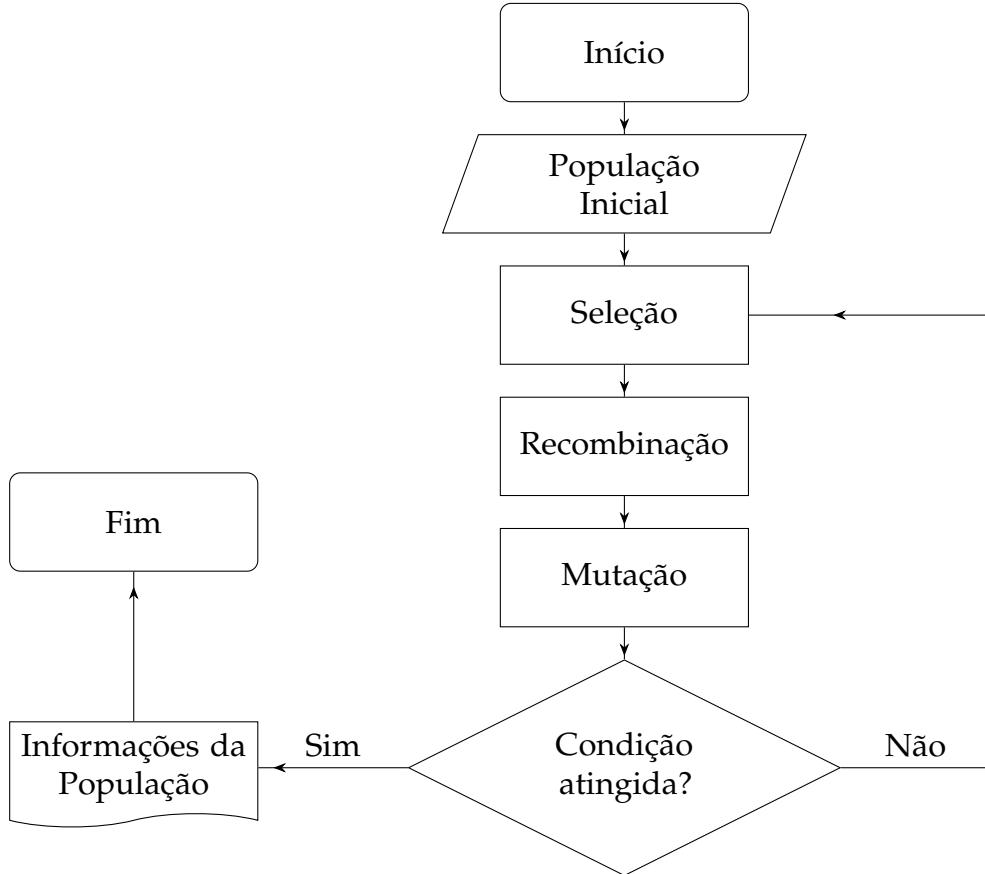


Figura 2 – Fluxograma geral de um algoritmo genético.

de problema de otimização proposto, essas etapas são presentes em todo algoritmo genético, e são bem resumidas no fluxograma presente na Figura 2.

1.3 Um Exemplo Físico: Ajuste das Estruturas de Banda de Dicalcogenetos de Metais de Transição

Dicalcogenetos de metais de transição (TMDC, do inglês *transition-metal dichalcogenides*) são materiais cuja composição é descrita por MX_2 , sendo M um metal de transição e X um calcogênio (S, Se ou Te). São conhecidos cerca de 60 compostos de tal classe, dos quais cerca de 20 apresentam estrutura cristalina na forma de finas camadas, com espessuras na ordem de 10 Å [3]. Por essa razão, podem ser categorizados como materiais bidimensionais.

A maioria desses materiais são sintéticos, mas alguns podem ser encontrados na natureza, como MoS_2 —dissulfeto de molibdênio—que ocorre naturalmente na forma do mineral molibdenita, sendo esse a principal fonte de obtenção do molibdênio. Uma fotografia desse mineral, bem como uma representação de sua estrutura cristalina são ilustrados na Figura 3.

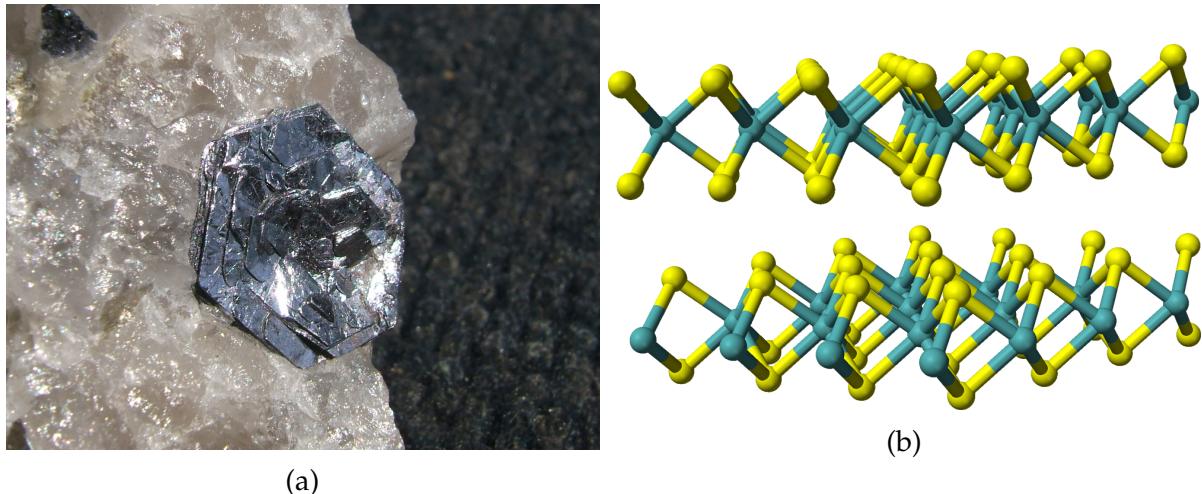


Figura 3 – (a) Molibdenita incrustada em uma massa de quartzo. (b) Representação da estrutura cristalina da molibdenita. Molibdênio em azul, enxofre em amarelo.

Esses materiais voltaram a ser amplamente estudados com a descoberta recente das propriedades eletrônicas do grafeno, pela qual K. S. Novoselov e A. K. Geim foram condecorados com o prêmio Nobel em 2010 [3]. Assim como no caso do grafeno, alguns TMDCs são semicondutores, característica essa que possibilita um grande leque de aplicações na indústria.

Dentre tais aplicações se destacam a manufatura de transistores, circuitos lógicos e amplificadores. Ademais, a estrutura eletrônica de alguns TMDCs torna propícia a formação de excitons em sua estrutura cristalina, o que abre margem para aplicações em dispositivos optoeletrônicos, como fotodetectores e células solares.

Em todos esses casos é de fundamental importância que se tenha conhecimento acerca dos níveis de energia do sólido em questão. Em particular, é de interesse prático o cálculo dos níveis de energia das bandas de valência e de condução — último nível de energia ocupado e primeiro nível de energia não ocupado, respectivamente.

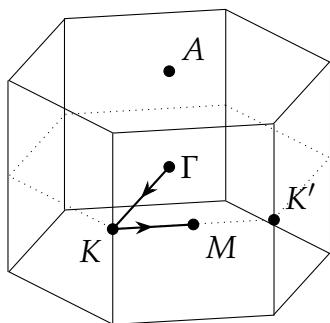


Figura 4 – Zona de Brillouin para monocamadas de TMDCs e principais pontos de simetria. Geralmente as bandas de valência e de condução são calculadas ao longo desses pontos, como no caminho exemplificado.

Desses valores se calcula a energia necessária para a formação de excitons[§], o que é determinante na confecção dos dispositivos optoeletrônicos enumerados no parágrafo anterior. Esse valor é conhecido como o *bandgap* do material.

Existe uma área na engenharia — denominada engenharia de *gap* — dedicada à manufatura de materiais com propriedades eletrônicas específicas, as quais são obtidas por meio do controle deste valor por intermédio de diversas técnicas. Um exemplo é o *doping*, que consiste a introdução de átomos distintos à composição do material em sua rede cristalina, aplicada com frequência na fabricação de diodos e transistores usados nos mais diversos dispositivos eletrônicos.

Usualmente, calcula-se tais bandas ao longo da primeira zona de Brillouin do cristal, a qual é a correspondente da célula primitiva no espaço recíproco[¶]. Para o caso dos matérias em questão, a zona de Brillouin é do tipo hexagonal, e se encontra ilustrada na Figura 4, conjuntamente aos principais pontos de simetria. Desses pontos de alta simetria, K e K' coincidem com os mínimos locais do *bandgap* do sistema, e por isso são de maior relevância.

Um dos métodos mais utilizados para o cálculo das bandas é a Teoria do Funcional Densidade (DFT, do inglês *density-functional theory*), o qual é um modelo quântico computacional para a aproximação das soluções de sistemas quânticos com múltiplas partículas, amplamente utilizado na química e na física, particularmente em física do estado sólido e física da matéria condensada.

No entanto, existem outros modelos para a estimativa das bandas de condução e de valência, como o modelo $k \cdot p$ — também conhecido como modelo de baixas energias, descrito em [4] — o qual se propõe a aproximar essas bandas em torno de K ou K' .

Das vantagens do uso desse modelo pode-se destacar o fato de ser um modelo aproximativo analítico, o que facilita o desenvolvimento de uma intuição física acerca do material, possibilitando a observação do comportamento das bandas de energia com a variação dos parâmetros que as descrevem. Também trata-se de um modelo computacionalmente barato, possibilitando a aproximação de sistemas que seriam computacionalmente complexos quando estimados com o uso do método DFT. Outra vantagem que será explorada subsequentemente na seção 3.2 é a possibilidade da inclusão de um campo magnético normal à monocamada do TMDC à descrição do sistema graças à existência de uma forma analítica para a hamiltoniana do cristal, o

[§]Um exciton é um par elétron-buraco formado pela transição de um elétron de valência para uma camada de condução. Em alguns sólidos e moléculas pode haver o transporte de excitons, fenômeno explorado semicondutores no geral.

[¶]Resumidamente, o espaço recíproco pode ser visto como o espaço gerado pela transformada de Fourier dos vetores primitivos, os quais constituem a chamada célula unitária, e por conseguinte, descrevem a estrutura cristalina de um sólido.

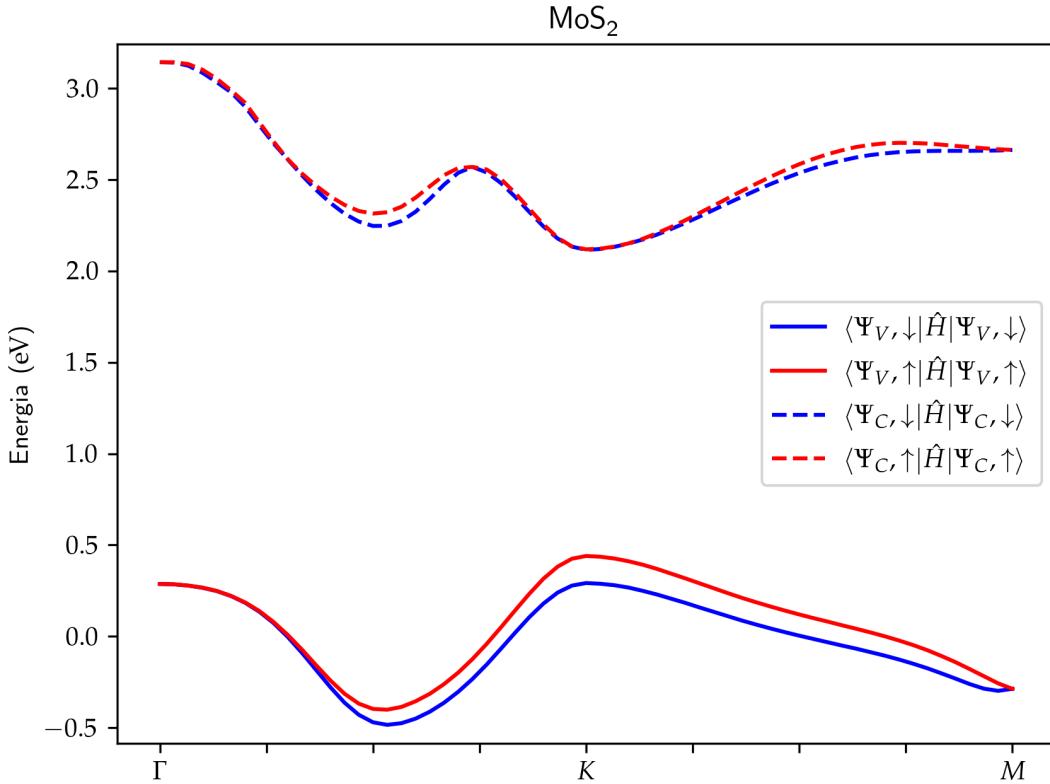


Figura 5 – Energias das bandas de condução e de valência ao longo do caminho ilustrado na Figura 4 para o dissulfeto de molibdênio.

que tornaria impraticável o uso do método DFT.

No modelo $k \cdot p$ a hamiltoniana na representação da base

$$\{|\Psi_C, \uparrow\rangle; |\Psi_C, \downarrow\rangle; |\Psi_V, \uparrow\rangle; |\Psi_V, \downarrow\rangle\} \quad (1.2)$$

é escrita na forma

$$\hat{H}_{kp}(\mathbf{k}) = \hat{H}_0 + \sum_{i=1}^3 \hat{H}_{kp}^{(i)}(\mathbf{k}) \quad (1.3)$$

onde Ψ_C e Ψ_V são as funções de onda do sistema com o elétron de valência em sua camada original e na camada de condução, respectivamente e $\hat{H}_{kp}^{(i)}(\mathbf{k})$ corresponde ao termo de ordem i com respeito a $\mathbf{k} = (k_x, k_y, k_z)$, um ponto do espaço recíproco.

Normalmente o vetor \mathbf{k} é escrito em uma base $\{\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z\}$ tal que Γ se encontra na origem, $\mathbf{e}_2 = \mathbf{K}'/|\mathbf{K}'|$, $\mathbf{e}_3 = \mathbf{A}/|\mathbf{A}|$ e $\mathbf{e}_1 = \mathbf{e}_2 \times \mathbf{e}_3$, sendo \mathbf{K}' e \mathbf{A} os vetores que localizam os pontos de simetria K' e A , respectivamente. No entanto, nesse modelo é necessário aplicar uma translação em tal base tal que a origem passe a ser localizada em K .

Os termos da expansão da hamiltoniana em função de \mathbf{k} são dados por

$$\begin{aligned} \hat{H}_0 &= \begin{pmatrix} E_F + \Delta - \tau\lambda_c & 0 & 0 & 0 \\ 0 & E_F + \tau\eta\lambda_v & 0 & 0 \\ 0 & 0 & E_F + \Delta + \tau\lambda_c & 0 \\ 0 & 0 & 0 & E_F - \tau\eta\lambda_v \end{pmatrix} \\ \hat{H}_{kp}^{(1)}(\mathbf{k}) &= \begin{pmatrix} 0 & \gamma_0 f_1(\mathbf{k}, \tau) & 0 & 0 \\ \gamma_0 f_1^\dagger(\mathbf{k}, \tau) & 0 & 0 & 0 \\ 0 & 0 & 0 & \gamma_0 f_1(\mathbf{k}, \tau) \\ 0 & 0 & \gamma_0 f_1^\dagger(\mathbf{k}, \tau) & 0 \end{pmatrix} \\ \hat{H}_{kp}^{(2)}(\mathbf{k}) &= \begin{pmatrix} \gamma_1 f_2(\mathbf{k}) & \gamma_3 f_3(\mathbf{k}, \tau) & 0 & 0 \\ \gamma_3 f_3^\dagger(\mathbf{k}, \tau) & \gamma_2 f_2(\mathbf{k}) & 0 & 0 \\ 0 & 0 & \gamma_1 f_2(\mathbf{k}) & \gamma_3 f_3(\mathbf{k}, \tau) \\ 0 & 0 & \gamma_3 f_3^\dagger(\mathbf{k}, \tau) & \gamma_2 f_2(\mathbf{k}) \end{pmatrix} \\ \hat{H}_{kp}^{(3)}(\mathbf{k}) &= \begin{pmatrix} \gamma_4 f_4(\mathbf{k}) & \gamma_6 f_5(\mathbf{k}, \tau) & 0 & 0 \\ \gamma_6 f_5^\dagger(\mathbf{k}, \tau) & \gamma_5 f_4(\mathbf{k}) & 0 & 0 \\ 0 & 0 & \gamma_4 f_4(\mathbf{k}) & \gamma_6 f_5(\mathbf{k}, \tau) \\ 0 & 0 & \gamma_6 f_5^\dagger(\mathbf{k}, \tau) & \gamma_5 f_4(\mathbf{k}) \end{pmatrix} \end{aligned} \quad (1.4)$$

sendo E_F a energia do nível de Fermi; Δ o *bandgap* do sistema; λ_c e λ_v os termos de acoplamento spin-órbita nos níveis de condução e valência, respectivamente; η um índice referente ao metal de transição em questão; τ o índice de vale, sendo $\tau = 1$ correspondendo ao vale em K , e $\tau = -1$ correspondendo ao vale em K' e γ_i são parâmetros de energia, os quais governam propriedades óticas do cristal. As funções f_i são dadas pelas expressões

$$\begin{aligned} f_1(\mathbf{k}, \tau) &= a(\tau k_x - ik_y) \\ f_2(\mathbf{k}) &= a^2(k_x^2 + k_y^2) \\ f_3(\mathbf{k}, \tau) &= a^2(\tau k_x + ik_y)^2 \\ f_4(\mathbf{k}, \tau) &= a^3 \tau k_x (k_x^2 - 3k_y^2) \\ f_5(\mathbf{k}, \tau) &= a^3 (k_x^2 + k_y^2)(\tau k_x - ik_y) \end{aligned} \quad (1.5)$$

onde a corresponde à constante de rede do respectivo sistema cristalino.

Portanto, uma vez conhecidos os valores para os parâmetros descritos, as energias dos estados eletrônicos que compõem a base na Equação 1.2 são obtidas pelo cálculo dos autovalores de \hat{H}_{kp} . Mas por vezes os valores dos parâmetros são desconhecidos, e, em posse das bandas de energia ao longo de um determinado caminho na zona de Brillouin, deseja-se obtê-los. Dados os autovalores de energia em um determinado caminho da zona de Brillouin como encontrar \hat{H}_{kp} correspondente? Esse é justamente um caso em que um algoritmo do tipo genético pode se mostrar útil.

Uma possível estratégia é tomar como espaço de busca uma região de \mathbb{R}^{11} —uma vez que serão tratados como parâmetros de ajuste E_F , Δ , λ_c , λ_v e γ_i —e como função objetivo

$$f(E_F, \Delta, \lambda_c, \lambda_v, \gamma_i) = \frac{1}{4N} \sum_{k=1}^N \sum_{i=1}^4 \left(E_{ik}^{(dft)} - E_{ik}^{(kp)} \right)^2 \quad (1.6)$$

onde $E_{ik}^{(dft)}$ corresponde ao i -ésimo maior valor de energia calculado no k -ésimo ponto de uma partição de tamanho N de um caminho escolhido, $E_{ik}^{(kp)}$ é o i -ésimo maior autovalor de \hat{H}_{kp} . Tal soma não precisa abarcar todos os pontos conhecidos pra um caminho em particular: uma vez que o modelo só se aplica em uma vizinhança de K , pode-se restringir a soma à um conjunto de pontos em torno de K .

Essa função nada mais é do que o desvio quadrático médio entre os dados conhecidos e os resultados de energia obtidos por meio do modelo, e espera-se que sua minimização com respeito aos parâmetros E_F , Δ , λ_c , λ_v e γ_i ajustem bem as bandas de energia conhecidas em torno de K .

Como para calcular o valor de $f(E_F, \Delta, \lambda_c, \lambda_v, \gamma_i)$ é necessário obter e ordenar os autovalores de \hat{H}_{kp} , qualquer método do tipo *hill climbing* não se aplica para a solução do problema. Além disso, nesse problema o espaço de busca tem uma dimensão considerável, o que também sugere o uso de métodos estocásticos como um algoritmo do tipo genético.

A eficácia de um algoritmo genético em um problema como este será posta a prova na Seção 3.2, onde será feito o ajuste da Hamiltoniana do modelo $k \cdot p$ para os cristais CrS₂ e CrSe₂. Os resultados serão comparados com os obtidos por outro método estocástico *Dual Annealing*, também muito utilizado em problemas de otimização similares.

1.4 A Implementação do Algoritmo Genético

A linguagem de programação escolhida para a implementação do algoritmo foi Python¹¹. Amplamente utilizada no meio acadêmico, trata-se de uma linguagem de fácil compreensão e aprendizado, além de contar com diversas bibliotecas já implementadas com o objetivo de resolver problemas recorrentes.

As duas principais bibliotecas utilizadas neste projeto foram NumPy e Matplotlib. A primeira se trata de uma biblioteca para manipulação numérica e vetorizada de matrizes. É implementada na linguagem C, o que confere performance às operações

¹¹ Versão 3.10.6, com NumPy na versão 1.23.2, Pandas na versão 1.4.3, Scipy na versão 1.9.1 e Matplotlib na versão 3.5.3. Código da implementação do algoritmo genético disponível em <https://github.com/davifeliciano/num_opt_ga>. Código e dados utilizados no ajuste dos níveis de energia de TMDCs disponíveis em <https://github.com/davifeliciano/kp_model_fit>.

numéricas, mantendo a facilidade de uso e versatilidade características da linguagem Python. Já a segunda, se trata de uma biblioteca para a criação de gráficos matemáticos, que será de suma importância na ilustração de forma clara dos resultados obtidos.

Os detalhes acerca da codificação escolhida para o material genético dos indivíduos, bem como os métodos utilizados nas etapas de seleção, recombinação e mutação foram propostos em 2006 por Roncaratti, Gargano e Silva[5], e serão abordados de forma breve no capítulo seguinte.

No problema do ajuste das bandas de energia dos Dicalcogenetos de Metais de Transição foi utilizada a biblioteca Pandas para a leitura dos dados referentes às bandas de energia calculadas por meio do método DFT e para a gravação dos parâmetros ajustados em um arquivo de saída. A biblioteca Scipy também foi utilizada para comparar os resultados obtidos com o algoritmo genético implementado com o método *Dual Annealing*, citado na seção anterior.

2 Metodologia

Este capítulo tem por objetivo abordar os detalhes acerca do algoritmo genético desenvolvido. Aqui é definida a codificação utilizada para o material genético de um indivíduo de uma população em termos de sua posição no espaço de busca e são formalizados os processos de seleção e recombinação em função da codificação escondida. Por fim, é apresentada uma estratégia elitista que será utilizada para melhorar a eficácia do algoritmo.

2.1 Codificação

Seja uma população de n indivíduos representada pelo conjunto

$$A = \{A_1, \dots, A_k, \dots, A_n\}, \quad (2.1)$$

onde cada indivíduo tem seu código genético representado por m cromossomos de l bits, de forma que, usando uma notação matricial,

$$A_k = \begin{bmatrix} a_{11}^{(k)} & \dots & a_{1m}^{(k)} \\ \vdots & \ddots & \vdots \\ a_{l1}^{(k)} & \dots & a_{lm}^{(k)} \end{bmatrix}, \quad (2.2)$$

onde $a_{ij}^{(k)} \in \{0, 1\}$, para i, j e k inteiros tais que $i \in [0, l]$, $j \in [0, m]$ e $k \in [0, n]$.

Para um problema de otimização numérica de uma função $f : \mathcal{C} \subseteq \mathbb{R}^m \rightarrow \mathbb{R}$, deve existir um mapa injetivo entre A e o conjunto

$$X = \{X_1, \dots, X_k, \dots, X_n\}, \quad (2.3)$$

onde

$$X_k = (x_1^{(k)}, \dots, x_j^{(k)}, \dots, x_m^{(k)}) \quad (2.4)$$

são vetores de \mathbb{R}^m e representam a posição do respectivo indivíduo no espaço de busca. Tal espaço é um subconjunto de \mathcal{C} definido pelo produto cartesiano

$$\mathcal{S} = \bigtimes_{j=1}^m [x_j^{(min)}, x_j^{(max)}], \quad (2.5)$$

sendo $x_j^{(min)}$ e $x_j^{(max)}$, respectivamente, os valores mínimo e máximo para a coordenada $x_j^{(k)}$.

Como l bits são capazes de representar números inteiros em $[0, 2^l - 1]$, uma forma natural de mapear tal intervalo em $[x_j^{(min)}, x_j^{(max)}]$ é

$$x_j^{(k)} = x_j^{(min)} + \frac{x_j^{(max)} - x_j^{(min)}}{2^l - 1} \sum_{i=1}^l a_{ij}^{(k)} 2^{i-1}, \quad (2.6)$$

cuja relação inversa é determinada por

$$\sum_{i=1}^l a_{ij}^{(k)} 2^{i-1} = \left\lfloor \frac{(x_j^{(k)} - x_j^{(min)})(2^l - 1)}{x_j^{(max)} - x_j^{(min)}} \right\rfloor. \quad (2.7)$$

Em suma, o j -ésimo cromossomo corresponde à representação binária do inteiro no segundo membro da Equação 2.7.

2.2 Seleção

Para o processo de seleção é introduzida uma nova função, denominada função desempenho. Tal função é alguma função $g : \mathbb{R} \rightarrow \mathbb{R}$ a partir da qual se calcula a probabilidade de seleção do indivíduo k

$$P_k = \frac{g(f(X_k))}{\sum_{j=1}^n g(f(X_j))}. \quad (2.8)$$

Existe uma liberdade para a escolha da função desempenho, desde que se respeite a condição a função seja sempre positiva. A ideia é que se escolha uma função desempenho que selecione os melhores indivíduos, mas ainda mantenha uma boa variedade genética na próxima geração. Assim, em toda geração, em média, haverá indivíduos em todo o espaço de busca, evitando que a solução convirja de forma prematura*.

Nesse trabalho a função desempenho usada foi um escalamento linear [1] $g(x) = ax + b$ sendo a e b definidos por

$$a = \begin{cases} \frac{\mu(h-1)}{f_{max}-\mu}, & \text{se } f_{min} > \frac{h\mu-f_{max}}{h-1} \\ \frac{\mu}{\mu-f_{min}}, & \text{caso contrário} \end{cases} \quad (2.9)$$

e

$$b = \begin{cases} \frac{\mu(f_{max}-h\mu)}{f_{max}-\mu}, & \text{se } f_{min} > \frac{h\mu-f_{max}}{h-1} \\ -\frac{\mu f_{min}}{\mu-f_{min}}, & \text{caso contrário} \end{cases}, \quad (2.10)$$

*Retomando a função usada como exemplo no capítulo anterior, definida na Equação 1.1, se escolhermos uma função de desempenho que privilegie de forma desproporcional pontos onde o valor da função é maior. Dependendo da distribuição inicial dos indivíduos, é possível que após algumas gerações, toda a população se concentre em $r = 1/2n$, que é um máximo local.

onde

$$f_{\min} = \min_{X_k \in X} f(X_k) , \quad (2.11)$$

$$f_{\max} = \max_{X_k \in X} f(X_k) , \quad (2.12)$$

h é um parâmetro real maior que 1 e μ é a média do valor de f sobre a população, dada por

$$\mu = \sum_{k=1}^n \frac{f(X_k)}{n} . \quad (2.13)$$

A seleção dos indivíduos que serão recombinados para formar a geração seguinte é feita por meio de uma roleta simples, permitindo ou não repetições. São feitos $n/2$ sorteios, já que para cada par de indivíduos recombinados, dois novos serão gerados[†]. Desse modo, a geração subsequente será constituída pelos indivíduos pais selecionados e os seus respectivos filhos, seguindo com n indivíduos.

2.3 Recombinação e Mutação

Seja o conjunto de indivíduos selecionados

$$S = \{S_1, \dots, S_k, \dots, S_{n/2}\} . \quad (2.14)$$

Estes indivíduos passarão, em pares adjacentes[‡], pelo processo de recombinação e mutação, para dar origem a geração seguinte da população. Esse processo é iniciado com o sorteio de duas posições em cada cromossomo, para cada um desses pares. Essas posições são chamadas de pontos de recombinação.

Em seguida, o material genético dos indivíduos filhos será obtido pela união das partes complementares de cada par de cromossomos dos pais e, por fim, são introduzidas mutações nos genes, com uma probabilidade predefinida.

Matematicamente, podemos realizar esse processo para cada par da seguinte forma. Primeiro, devemos sortear os pontos de recombinação. Esse processo é feito por meio da geração de m pares de números naturais aleatórios distintos $\alpha_j^{(k)}$ e $\beta_j^{(k)}$ tais que $0 \leq \alpha_j^{(k)} < \beta_j^{(k)}$ e $\alpha_j^{(k)} \leq \beta_j^{(k)} < l$ onde $j = 0, 1, \dots, m$ e $k = 0, 2, 4, \dots, n/2 - 2$. Depois, é montada uma matriz R denominada máscara de recombinação, definida por

$$r_{ij}^{(k)} = \begin{cases} 0, & \text{se } i \leq \alpha_j^{(k)} \text{ ou } i > \beta_j^{(k)} \\ 1, & \text{caso contrário} \end{cases} , \quad (2.15)$$

[†]Por esse motivo, n deve ser um múltiplo de 4.

[‡]O par (S_1, S_2) dá origem a dois indivíduos. O par (S_3, S_4) forma outros dois, e assim segue até que $n/2$ novos indivíduos sejam gerados.

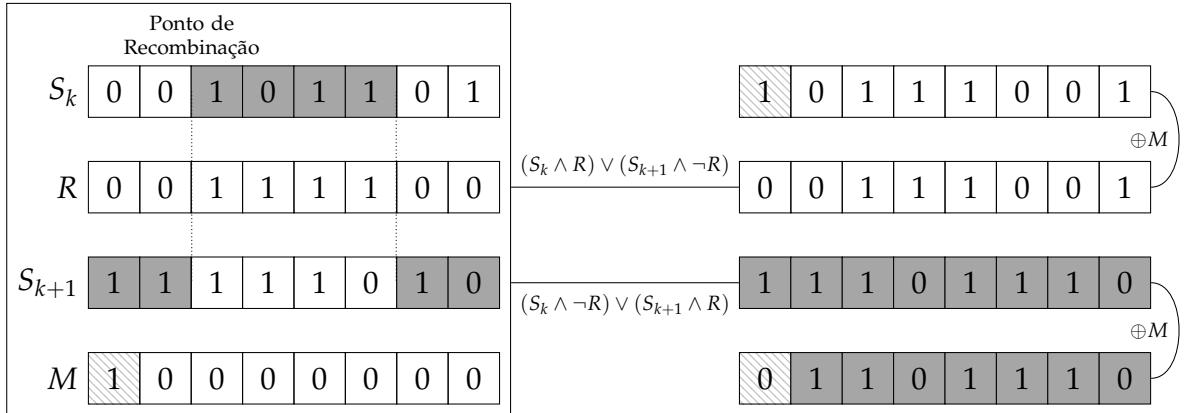


Figura 6 – Diagrama ilustrando — em um caso com $m = 1$ e $l = 8$ — a recombinação entre os indivíduos S_k e S_{k+1} e mutação nos filhos gerados segundo as respectivas máscaras de recombinação e mutação R e M .

bem como uma máscara de mutação M , tal que

$$m_{ij}^{(k)} = \begin{cases} 1, & \text{se houver mutação} \\ 0, & \text{caso contrário} \end{cases}. \quad (2.16)$$

Finalmente, o conjunto dos indivíduos resultantes da recombinação da seleção no conjunto definido pela Equação 2.14 será

$$S' = \{S'_1, \dots, S'_k, \dots, S'_{n/2}\} \quad (2.17)$$

tal que

$$s'_{ij}^{(k)} = (s_{ij}^{(k)} \wedge r_{ij}^{(k)}) \vee (s_{ij}^{(k+1)} \wedge \neg r_{ij}^{(k)}) \oplus m_{ij}^{(k)} \quad (2.18)$$

e

$$s'_{ij}^{(k+1)} = (s_{ij}^{(k)} \wedge \neg r_{ij}^{(k)}) \vee (s_{ij}^{(k+1)} \wedge r_{ij}^{(k)}) \oplus m_{ij}^{(k)}, \quad (2.19)$$

onde \neg , \wedge , \vee e \oplus são os operadores lógicos de negação, conjunção, disjunção e disjunção exclusiva, respectivamente.

Ao final desse processo — resumido no diagrama presente na Figura 6 — teremos uma nova população, sobre a qual poderemos ordenar os indivíduos segundo a função desempenho, verificar as características dos primeiros, e, caso necessário, repetir o algoritmo desde a etapa de seleção.

2.4 Estratégia Elitista

A fim de garantir a reprodução das características dos melhores indivíduos na geração seguinte, uma estratégia elitista pode ser tomada [1] [5]. Seja S_e o melhor indivíduo[§] do conjunto S definido na Equação 2.14. A ideia dessa abordagem é

[§]Aquele cuja solução correspondente X_e satisfaz $g(f(X_e)) = \max_{X_k \in X} g(f(X_k))$.

introduzir — antes do processo se recombinação — cópias de S_ϵ na população[¶] com o objetivo de preservar seu material genético, assim como recombina-lo com o de outros indivíduos, na esperança de obter soluções ainda melhores para o problema.

Para isso, substituímos o conjunto S por

$$S = \left\{ \underbrace{S_\epsilon, \dots, S_\epsilon}_{e_1 \text{ cópias}}, \underbrace{S_\epsilon, \dots, S_\epsilon}_{e_2 \text{ cópias}}, \underbrace{S_{e_1+e_2+1}, \dots, S_{n/2}}_{e_3 \text{ cópias dentre os restantes}} \right\}, \quad (2.20)$$

onde os primeiros $e_1 + e_2$ indivíduos foram trocados por S_ϵ ^{||}, e, dentre os indivíduos restantes, são feitas e_3 substituições similares em posições aleatórias.

Ao final desse processo, prossegue-se para a etapa de recombinação, com o detalhe de que os primeiros e_1 indivíduos terão probabilidade de mutação $p_1 = 0$, os e_2 indivíduos seguintes e o restante da população terão probabilidades de mutação não nulas p_2 e p_3 . Como consequência, S_ϵ será parte da população seguinte, uma vez que a recombinação entre um indivíduo e seu clone dá origem a dois outros clones, na ausência de mutações. Ademais, cópias de S_ϵ com mutações em seu material genético também estarão presentes, bem como cruzamentos de S_ϵ com até outros e_3 indivíduos distintos.

[¶]O conjunto dessas cópias é chamado de elite da população.

^{||}Como a recombinação ocorre em pares, e_1 e e_2 devem ser pares.

3 Resultados

Neste capítulo são exibidos os resultados obtidos da aplicação do algoritmo desenvolvido na otimização de algumas funções. Na seção 3.1 o algoritmo é utilizado para acusar máximos de algumas funções de \mathbb{R}^2 em \mathbb{R} e na seção 3.2 é feito o ajuste das bandas de energia dos TMDCs CrS_2 e CrSe_2 minimizando o desvio quadrático médio entre os autovalores da hamiltoniana do modelo $k \cdot p$ e os respectivos níveis calculados previamente via DFT. Por fim, com os parâmetros ajustados, são ilustrados alguns resultados físicos relevantes.

3.1 Testes em Funções

Nessa seção são exibidos os resultados de alguns testes em funções reais em \mathbb{R}^2 . Em cada teste, 8 populações com 1000 indivíduos com cromossomos de 32 bits são evoluídas por 100 gerações. O espaço de busca considerado foi $[-1, 1] \times [-1, 1]$, onde os indivíduos foram distribuídos aleatoriamente. O valor usado para o parâmetro da função desempenho foi $h = 2$, as configurações para a elite foram $e_1 = 4$, $e_2 = 6$ e $e_3 = 10$ e as probabilidades de mutação p_2 e p_3 escolhidas foram $p_2 = 5\%$ e $p_3 = 5\%$.

Ao final do processo, foram gerados gráficos com as curvas de nível de cada função e com as posições de todos os integrantes de todas as populações. Os 8 melhores indivíduos são marcados de forma distinta, afim de atestar se o algoritmo foi capaz ou não de encontrar a solução real do problema. Um segundo gráfico com os valores das funções dos 200 melhores indivíduos de cada população no decorrer das gerações foi gerado, a fim de mostrar, em caso de convergência, sua rapidez.

Ambos os gráficos foram feitos novamente com probabilidades de mutação $p_2 = p_3 = 20\%$ para demonstrar o papel da mutação no decorrer do algoritmo, e averiguar seu impacto na diversidade genética dos indivíduos. Por fim, é feita uma breve discussão sobre a performance da implementação do algoritmo e sua complexidade de tempo de execução.

A primeira função testada foi

$$f_1(x, y) = \cos(9\pi r) \exp\left\{-\frac{r^2}{(0,4)^2}\right\}, \text{ com} \\ r = \sqrt{(x - 0,5)^2 + (y - 0,5)^2} \quad (3.1)$$

cujo gráfico se encontra na Figura 7a e os resultados obtidos estão dispostos nas

Figuras 8, 9, 10 e 11. A segunda função testada foi

$$f_2(x, y) = 0,8 \exp \left\{ -\frac{r_1^2}{(0,3)^2} \right\} + 0,88 \exp \left\{ -\frac{r_2^2}{(0,03)^2} \right\}, \text{ onde}$$

$$r_1 = \sqrt{(x - 0,5)^2 + (y - 0,5)^2} \text{ e}$$

$$r_2 = \sqrt{(x - 0,6)^2 + (y - 0,1)^2},$$
(3.2)

cujo gráfico se encontra na Figura 7b com os resultados obtidos ilustrados nas Figuras 12, 13, 14 e 15.

Como pode ser observado, em todos os casos, os melhores indivíduos foram localizados com precisão no máximo global. Não obstante, uma variedade genética proporcional a probabilidade de mutação foi mantida, dada uma escolha correta de p_2 e p_3 para cada problema. Nos casos em que $p_2 = 20\%$, os indivíduos, ao final do processo, se encontravam espalhados por todo o espaço de busca.

Vale ressaltar porém que o valor necessário de p_2 e p_3 para que uma população tenha a distribuição desejada depende da função a ser otimizada, como pode ser visto comparando as Figuras 10 e 11 com as Figuras 14 e 15. Assim, a influência dos parâmetros p_2 , p_3 , e_1 , e_2 e e_3 no comportamento da população deve ser estudada em cada caso, afim de extrair do algoritmo o resultado desejado.

Outra vantagem do algoritmo é que os máximos locais também puderam ser encontrados. Isso pode ser observado especialmente na Figura 11, onde visivelmente há uma concentração de indivíduos em $f_1(x, y) \approx 0,74$ e $f_1(x, y) \approx 0,29$, que correspondem aos dois primeiros máximos locais. Algo similar pode ser observado na segunda função, nas primeiras gerações da Figura 15, com alguma concentração da população em vermelho em $f_1(x, y) \approx 0,8$.

Claro que, mesmo nos casos em que não é possível inferir o máximo local da evolução de uma parcela da população — como ocorre na Figura 9 — uma análise estatística feita sobre os valores das funções desempenho dos indivíduos e suas respectivas posições ainda seria capaz de acusá-lo.

Em alguns casos, 100 gerações não são o suficiente para determinar a solução correta. Na população em vermelho na Figura 13 houve uma convergência rápida para o máximo local, onde todos os 200 melhores indivíduos permaneceram durante toda a duração do teste. Algo similar ocorreu na população em verde, para a maioria dos indivíduos da elite. Entretanto, o número de gerações sempre pode ser escolhido conforme a necessidade.

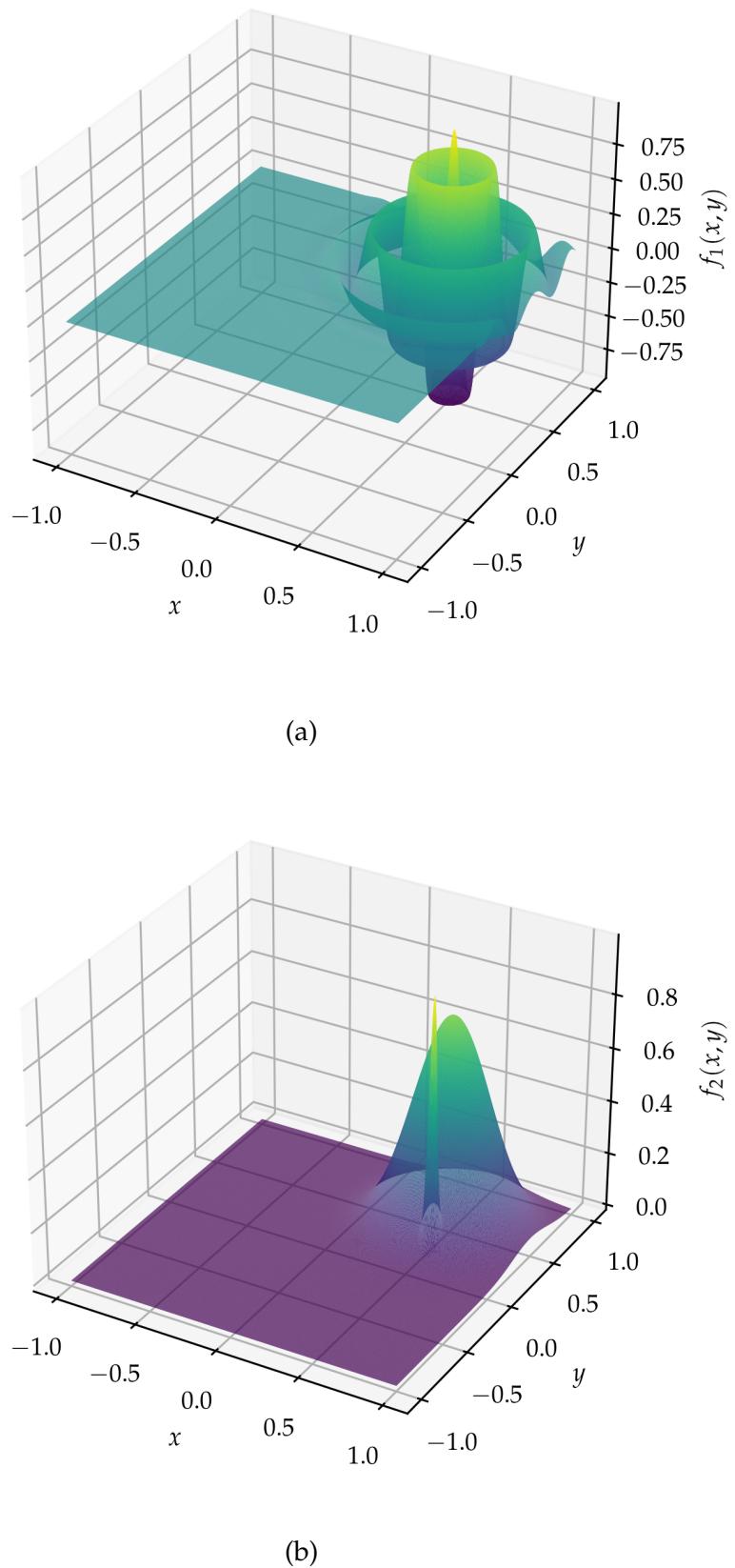


Figura 7 – (a) Gráfico da função $f_1(x, y)$. (b) Gráfico da função $f_2(x, y)$.

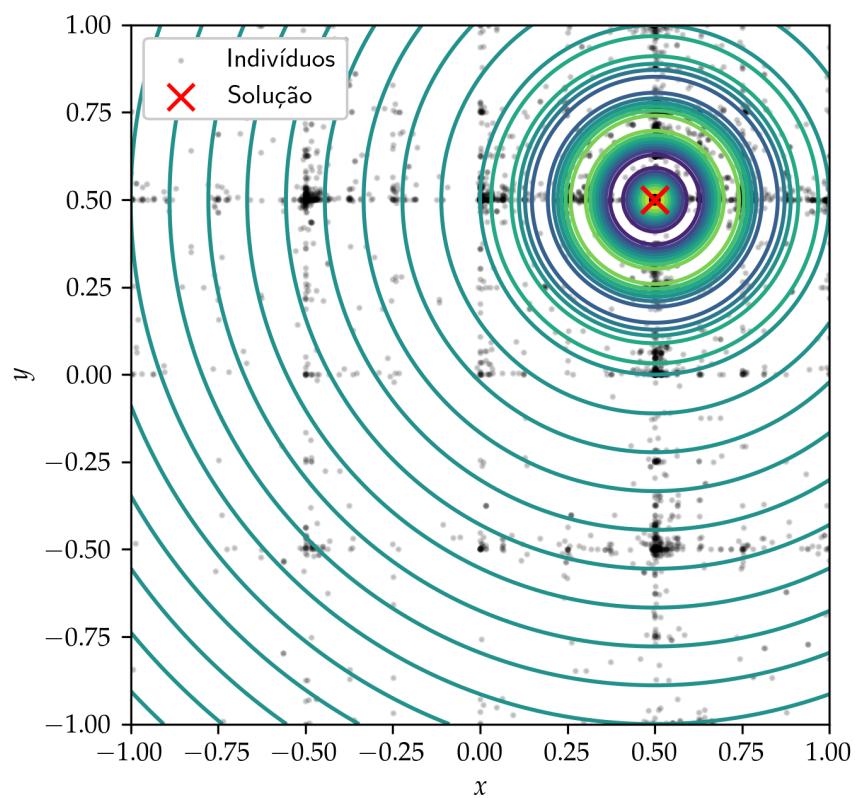


Figura 8 – Curvas de nível da função $f_1(x, y)$. Os pontos em preto indicam as posições dos indivíduos de 8 populações em sua 100^a geração na otimização da função, com $p_2 = p_3 = 5\%$. Marcado com um \times vermelho estão os melhores indivíduos de cada população.

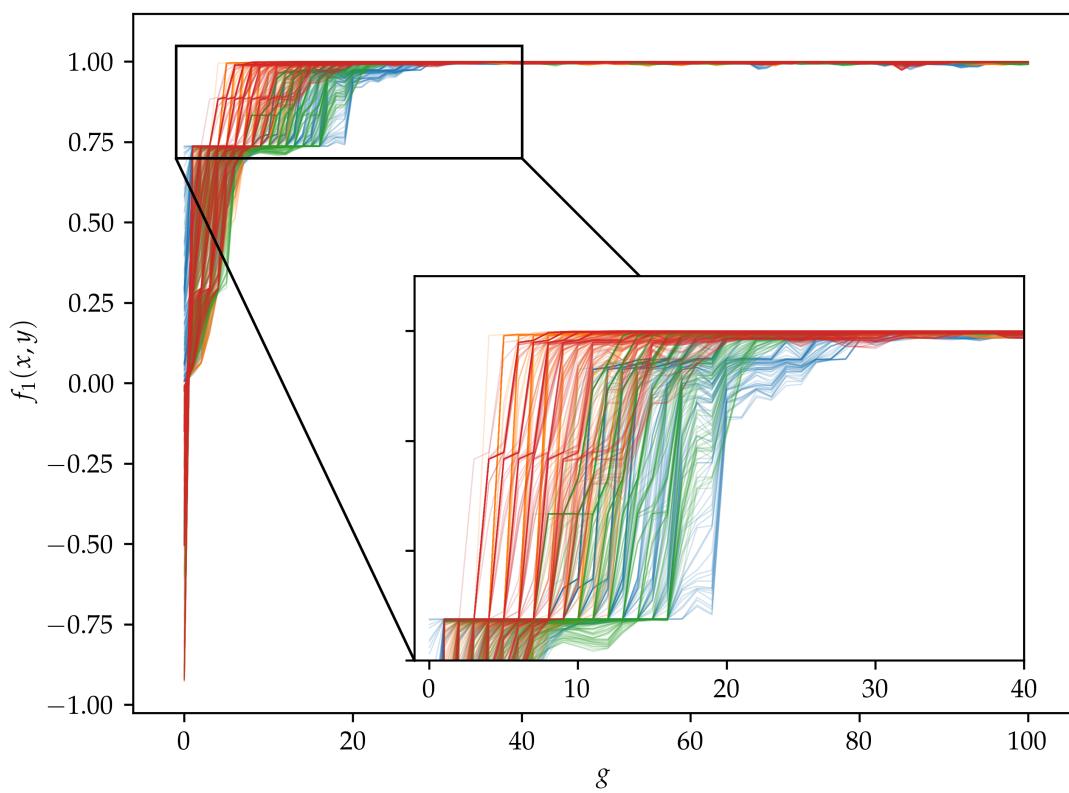


Figura 9 – Evolução dos valores da função $f_1(x, y)$ para os melhores 200 indivíduos de cada população, diferenciadas por cor, em termos da geração g , com $p_2 = p_3 = 5\%$.

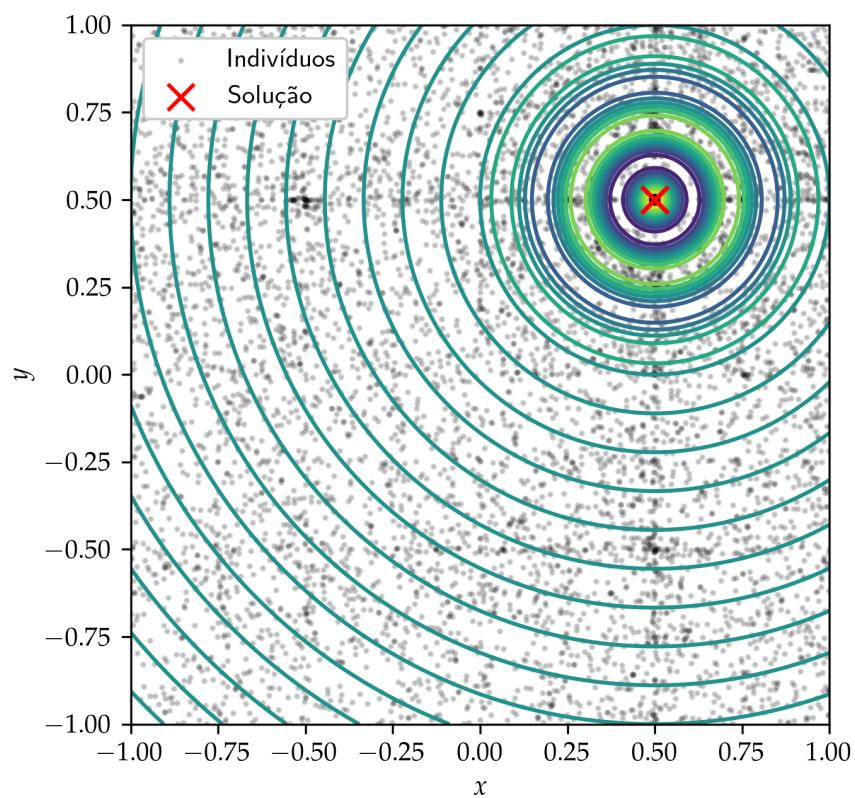


Figura 10 – Curvas de nível da função $f_1(x, y)$. Os pontos em preto indicam as posições dos indivíduos de 8 populações em sua 100^a geração na otimização da função, com $p_2 = p_3 = 20\%$. Marcado com um \times vermelho estão os melhores indivíduos de cada população.

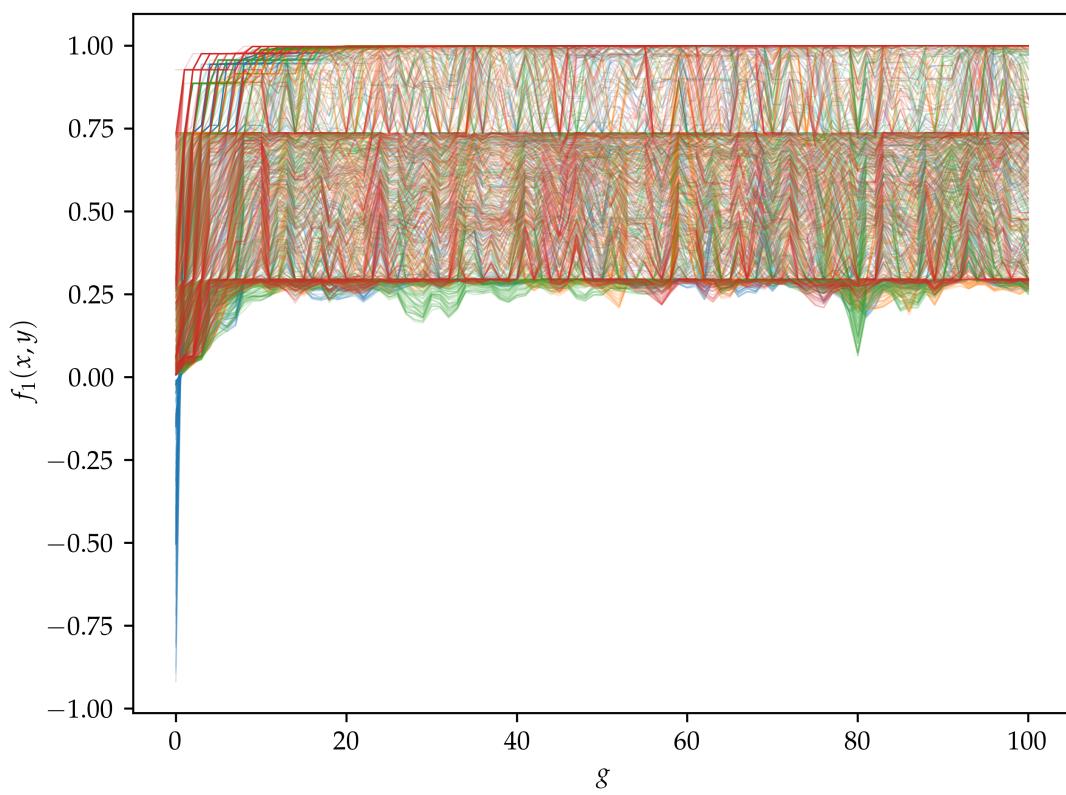


Figura 11 – Evolução dos valores da função $f_1(x, y)$ para os melhores 200 indivíduos de cada população, diferenciadas por cor, em termos da geração g , com $p_2 = p_3 = 20\%$.

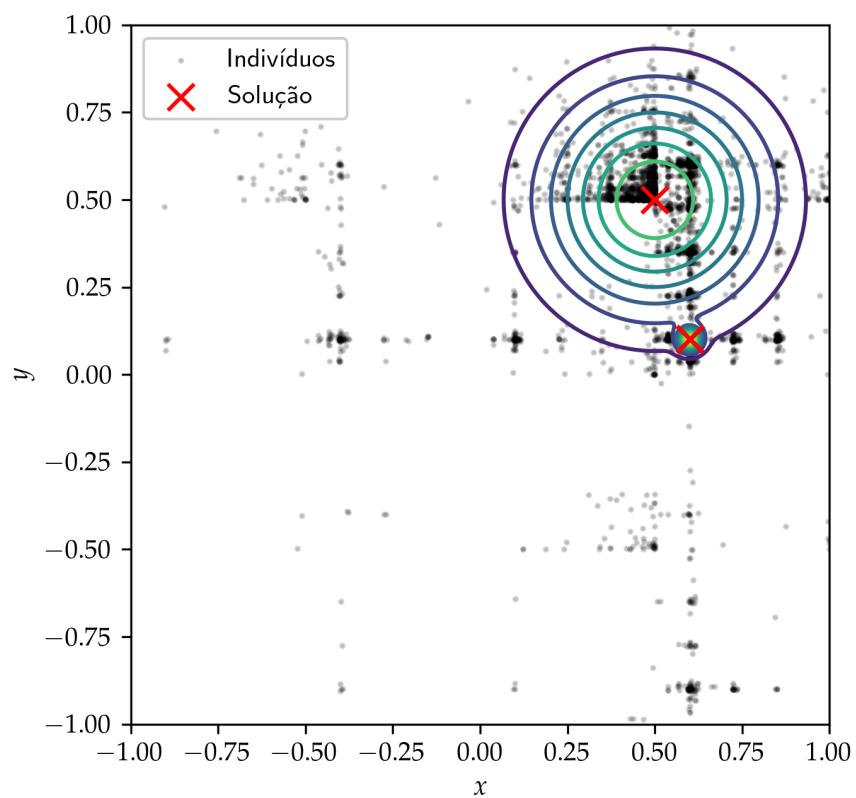


Figura 12 – Curvas de nível da função $f_2(x, y)$. Os pontos em preto indicam as posições dos indivíduos de 8 populações em sua 100^a geração na otimização da função, com $p_2 = p_3 = 5\%$. Marcado com um \times vermelho estão os melhores indivíduos de cada população.

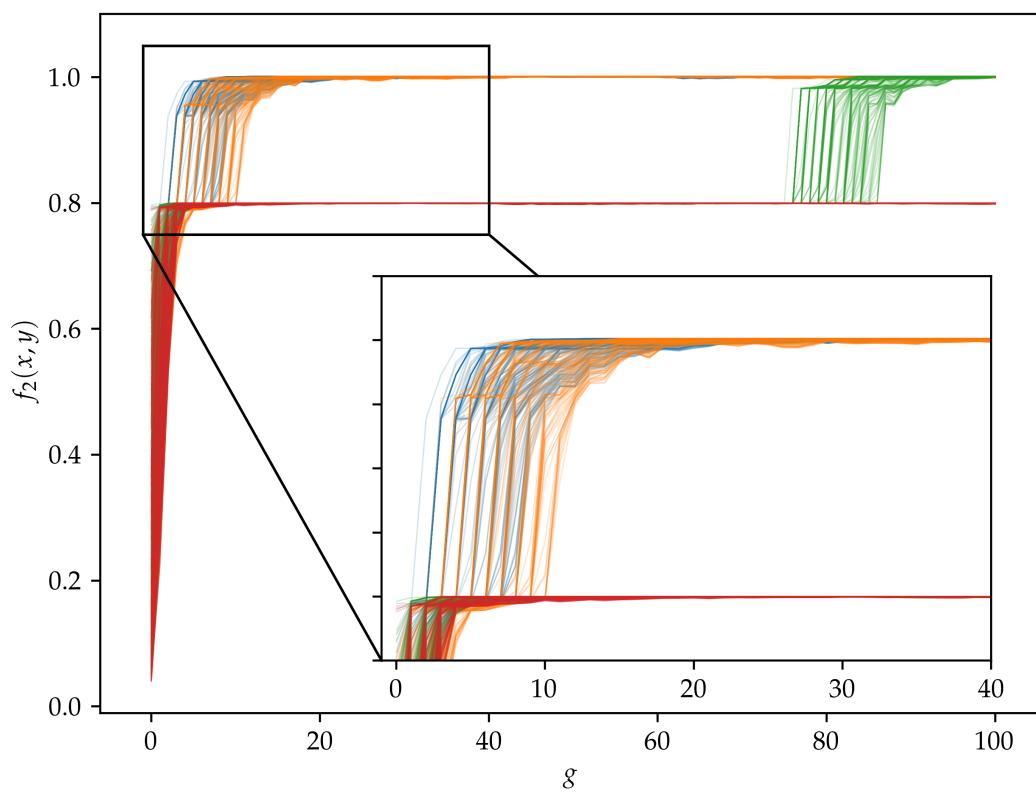


Figura 13 – Evolução dos valores da função $f_2(x,y)$ para os melhores 200 indivíduos de cada população, diferenciadas por cor, em termos da geração g , com $p_2 = p_3 = 5\%$.

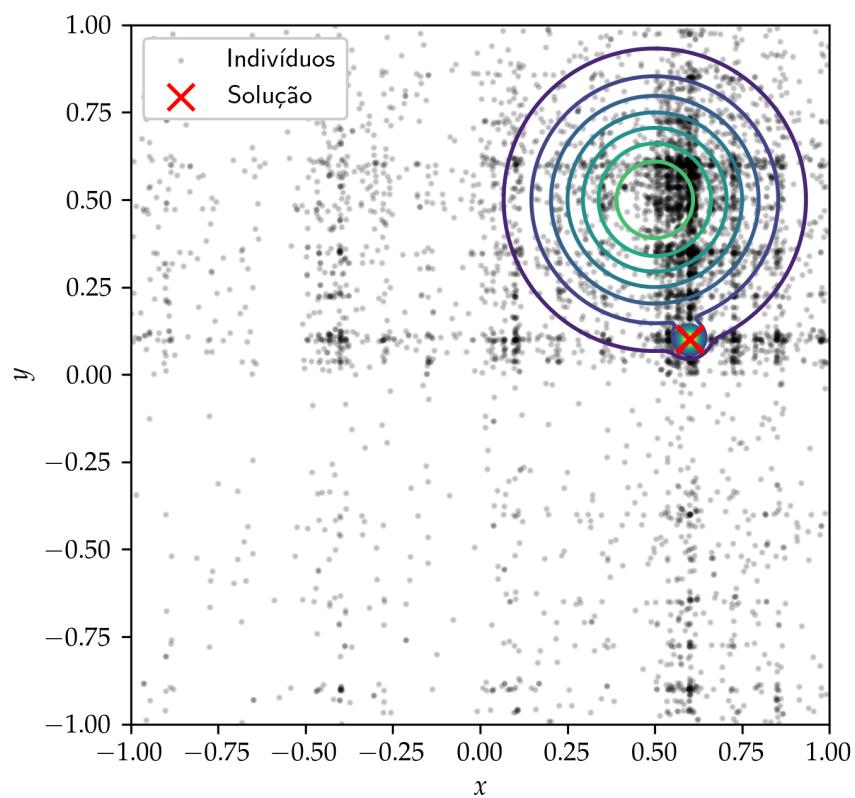


Figura 14 – Curvas de nível da função $f_2(x, y)$. Os pontos em preto indicam as posições dos indivíduos de 8 populações em sua 100^a geração na otimização da função, com $p_2 = p_3 = 20\%$. Marcado com um \times vermelho estão os melhores indivíduos de cada população.

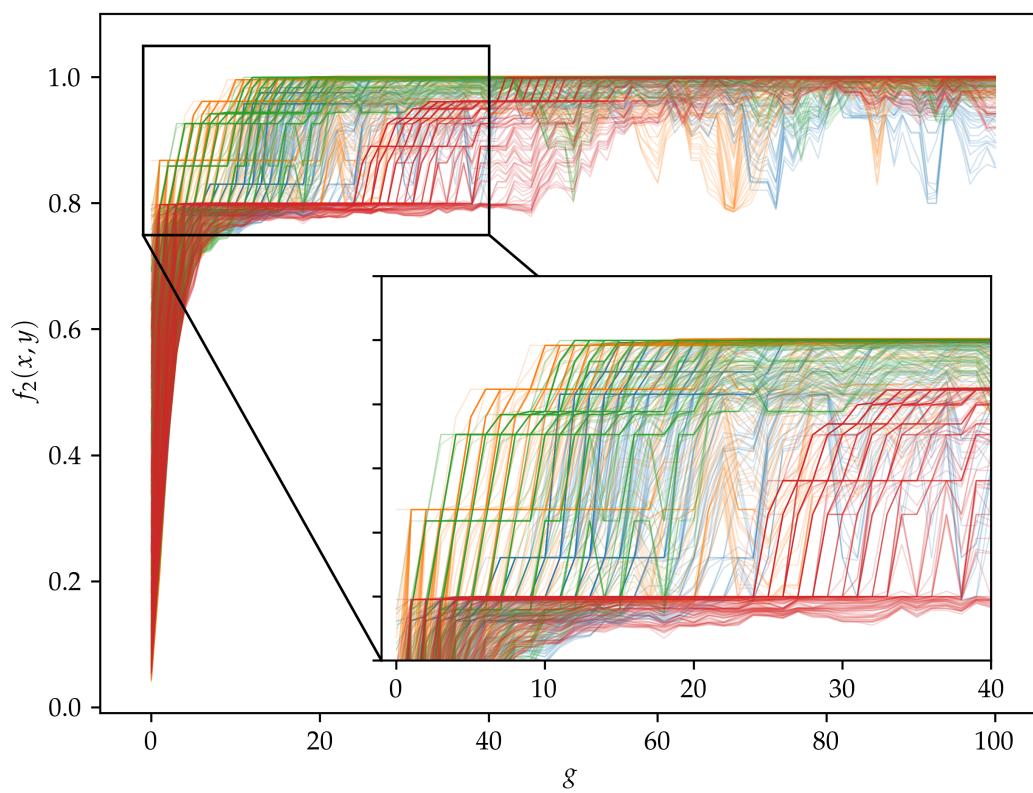


Figura 15 – Evolução dos valores da função $f_2(x,y)$ para os melhores 200 indivíduos de cada população, diferenciadas por cor, em termos da geração g , com $p_2 = p_3 = 20\%$.

3.2 Ajuste das Estruturas de Banda dos Cristais CrS_2 e CrSe_2

Para o ajuste das bandas de energia dos materiais CrS_2 e CrSe_2 foram executados 16 processos simultâneos com 1000 indivíduos cada, evoluídos no decorrer de 100 gerações, com probabilidades de mutação $p_2 = p_3 = 5\%$ e $h = 2$. A distribuição da elite nas populações ocorreu de forma que $e_1 = 5\%$, $e_2 = e_3 = 10\%$, sendo os percentuais tomados em relação ao tamanho da população. A região de busca considerada no processo de ajuste é dada pelo produto cartesiano dos intervalos na Tabela 1.

	$x_j^{(min)}$	$x_j^{(max)}$
E_F	-1,0	1,0
Δ	0,5	1,2
λ_c	0,0	1,0
λ_v	0,0	1,0
γ_i	-1,0	1,0

Tabela 1 – Intervalos de busca para cada parâmetro considerados no ajuste das bandas de energia via modelo $k \cdot p$. Definidos estes intervalos, a região de busca é calculada conforme a Equação 2.5.

Para fins de comparação, foi feito um ajuste da mesma função da Equação 1.6 porém via outro método denominado *Dual Annealing**. Trata-se de um método também estocástico, com enfase em busca local e global, assim como o Algoritmo Genético implementado. Para a busca, foi utilizado um número máximo de iterações de 2000, e uma temperatura inicial $T_{qv}(1) = 2,5 \times 10^4$. Quanto aos parâmetros relativos ao modelo $k \cdot p$, para CrS_2 e CrSe_2 , os parâmetros de rede considerados para os cálculos, bem como os valores esperados para Δ se encontram enumerados na Tabela 2. Para o metal de transição em questão, $\eta = 1$ e uma vez que a expansão é feita em torno do ponto do vale K , $\tau = 1$.

	CrS_2	CrSe_2
a (Å)	3,022302679	3,167287237
Δ (eV)	0,942	0,763

Tabela 2 – Valores para os parâmetros de rede considerados nos processos de otimização e valores esperados para o *bandgap*, para cada um dos materiais considerados.

Os gráficos com os resultados obtidos por meio dos dois métodos utilizados aplicados a cada um dos TMDCs se encontram nas Figuras 16 e 17, com os parâmetros ajustados detalhados nas Tabelas 3a e 3b. Observando esses gráficos, é possível atestar

*Documentado em <https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.dual_annealing.html>, com definição do modelo e respectivas referências enumeradas ao final da página.

a eficácia de ambos os métodos de ajuste, uma vez que as bandas de energia da expansão de 3^a ordem do modelo $k \cdot p$ praticamente coincidiram — na vizinhança de K — com as bandas calculadas pelo método DFT.

O *bandgap* estimado com ambos os métodos apresentou uma discrepância menor que 5% com relação ao valor esperado para a expansão de 3^a ordem, o que atesta a eficácia de ambos os métodos e do modelo $k \cdot p$ em si. Ademais, é possível observar que os valores obtidos para os parâmetros E_F , Δ , λ_c e λ_v e para o valor mínimo de f encontrado em ambas as ordens utilizadas não apresentaram grande discrepância quando comparados entre os dois métodos utilizados para a busca. Já o parâmetro γ_0 ajustado também apresentou uma variação muito pequena, porém com uma troca de sinal.

Dito isso, pode-se agora fazer uso prático dos parâmetros ajustados. Como enunciado na Seção 1.3, o modelo $k \cdot p$ possibilita a inclusão de um termo de interação referente a um campo magnético uniforme $\mathbf{B} = Be_z$ à hamiltoniana \hat{H}_{kp} . Assim, os novos níveis de energia das bandas de valência e de condução serão dados pelos novos autovalores. Fazendo isso para a expansão de 1^a ordem obteremos as energias

$$E_{\pm}(B, n) = \frac{\lambda_v \tau s_z}{2} \pm \sqrt{\frac{(\Delta - \lambda_v \tau s_z)^2}{4} + \frac{2\gamma_0^2 a^2 e B n}{\hbar}}, \quad (3.3)$$

onde os sinais + e - se referem, respectivamente, as bandas de condução e de valência, e $s_z = 1$ e $s_z = -1$ representam estados de spin *up* e *down*, e n é um natural não nulo [6]. Para o caso particular em que $n = 0$,

$$E(n = 0) = \begin{cases} -\Delta/2 + \lambda_v s_z, & \text{se } \tau = 1 \\ \Delta/2, & \text{se } \tau = -1 \end{cases} \quad (3.4)$$

correspondendo a níveis de valência e de condução.

Os gráficos das Figuras 18a e 18b ilustram o novo comportamento das bandas de condução e de valência no vale K para ambos os materiais estudados: com a presença de um campo externo, há a quebra de degenerescência dando origem a novos níveis de energia, conhecidos como níveis de Landau. De forma similar, nas Figuras 19a e 19b são retratados estes mesmos níveis para o vale K' .

	Algorítmo Genético		<i>Dual Annealing</i>	
	1 ^a Ordem	3 ^a Ordem	1 ^a Ordem	3 ^a Ordem
f	0,003858	0,000335	0,003860	0,000165
E_F	-0,052734	0,023191	-0,052109	0,010791
Δ	1,077500	0,954347	1,076284	0,976370
λ_c	-0,002289	0,006503	0,000000	0,000000
λ_v	0,031810	0,030079	0,032250	0,033154
γ_0	0,475147	0,508048	-0,475765	0,662441
γ_1		0,227823		0,038586
γ_2		-0,216329		-0,050962
γ_3		0,185349		0,196977
γ_4		-0,063952		-0,128763
γ_5		0,037538		0,116953
γ_6		-0,196858		-0,147924

(a)

	Algorítmo Genético		<i>Dual Annealing</i>	
	1 ^a Ordem	3 ^a Ordem	1 ^a Ordem	3 ^a Ordem
f	0,002695	0,000261	0,002695	0,000129
E_F	0,657808	0,722708	0,657704	0,715254
Δ	0,897031	0,792868	0,896809	0,814547
λ_c	0,006592	0,007456	0,006680	0,007329
λ_v	0,045311	0,047549	0,045330	0,042298
γ_0	-0,381442	-0,649511	0,381519	0,456269
γ_1		-0,090387		0,137704
γ_2		0,090028		-0,154747
γ_3		-0,075382		0,124848
γ_4		-0,059988		-0,030592
γ_5		0,037979		0,016737
γ_6		0,110312		-0,169387

(b)

Tabela 3 – Parâmetros da hamiltoniana \hat{H}_{kp} ajustados para CrS₂ (a) e para CrSe₂ (b) usando as expansões de 1^a e 3^a ordem de \hat{H}_{kp} , bem como os valores para a função objetivo f correspondente.

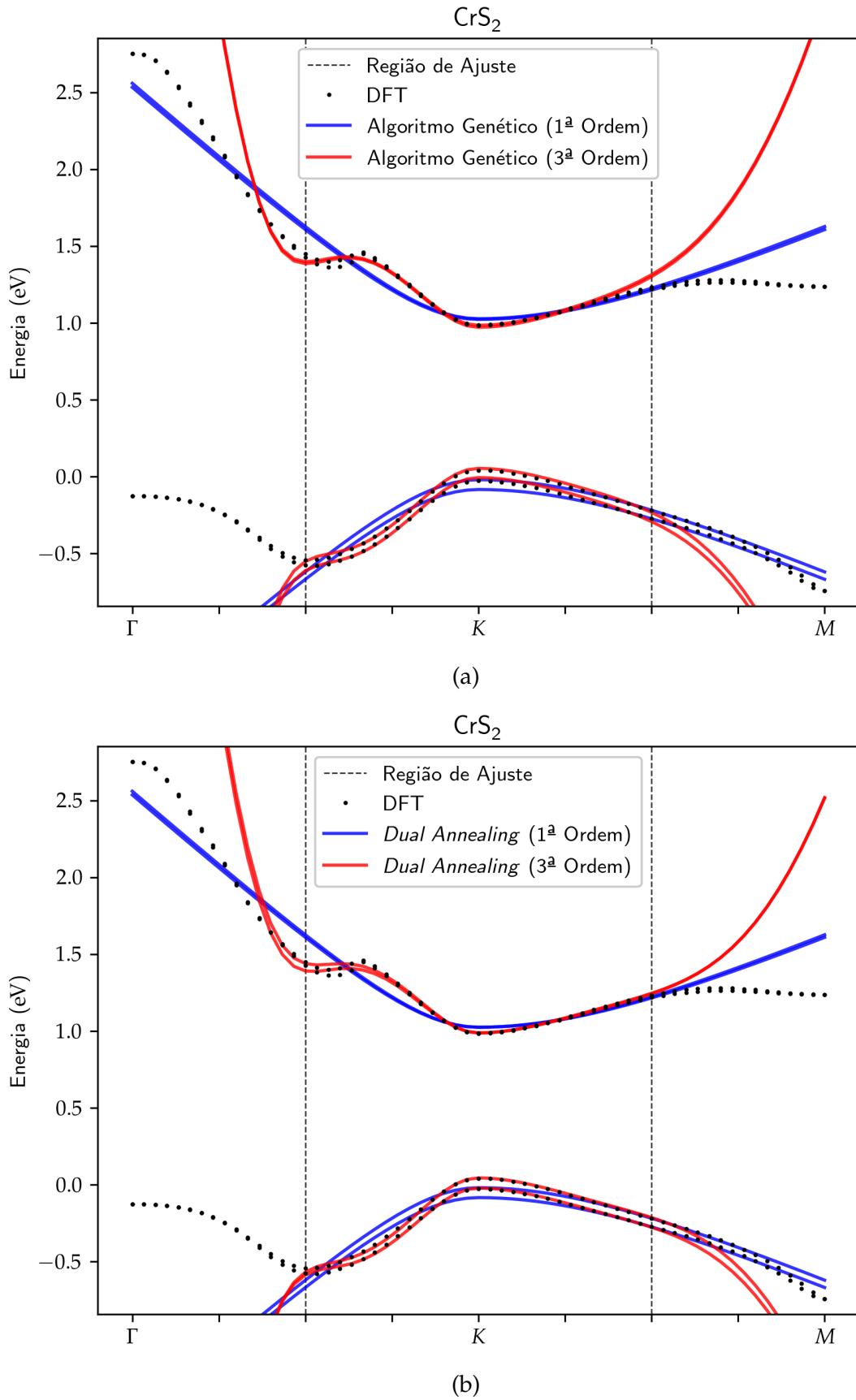


Figura 16 – Gráficos das bandas de energia ajustadas para CrS₂ via Algoritmo Genético (a) e via *Dual Annealing* (b) usando as expansões de 1^a e 3^a ordem de \hat{H}_{kp} .

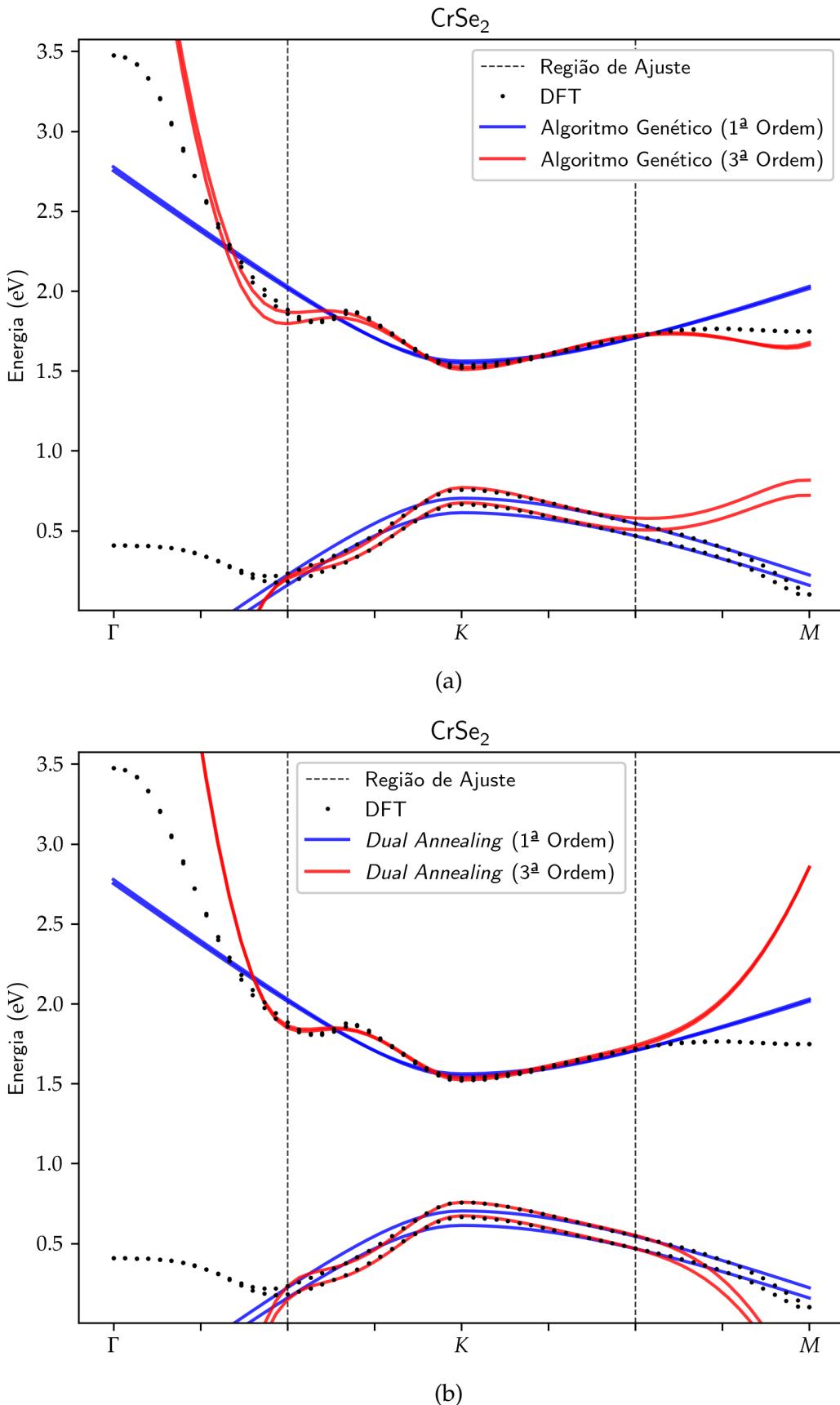


Figura 17 – Gráficos das bandas de energia ajustadas para CrSe₂ via Algoritmo Genético (a) e via Dual Annealing (b) usando as expansões de 1^a e 3^a ordem de \hat{H}_{kp} .

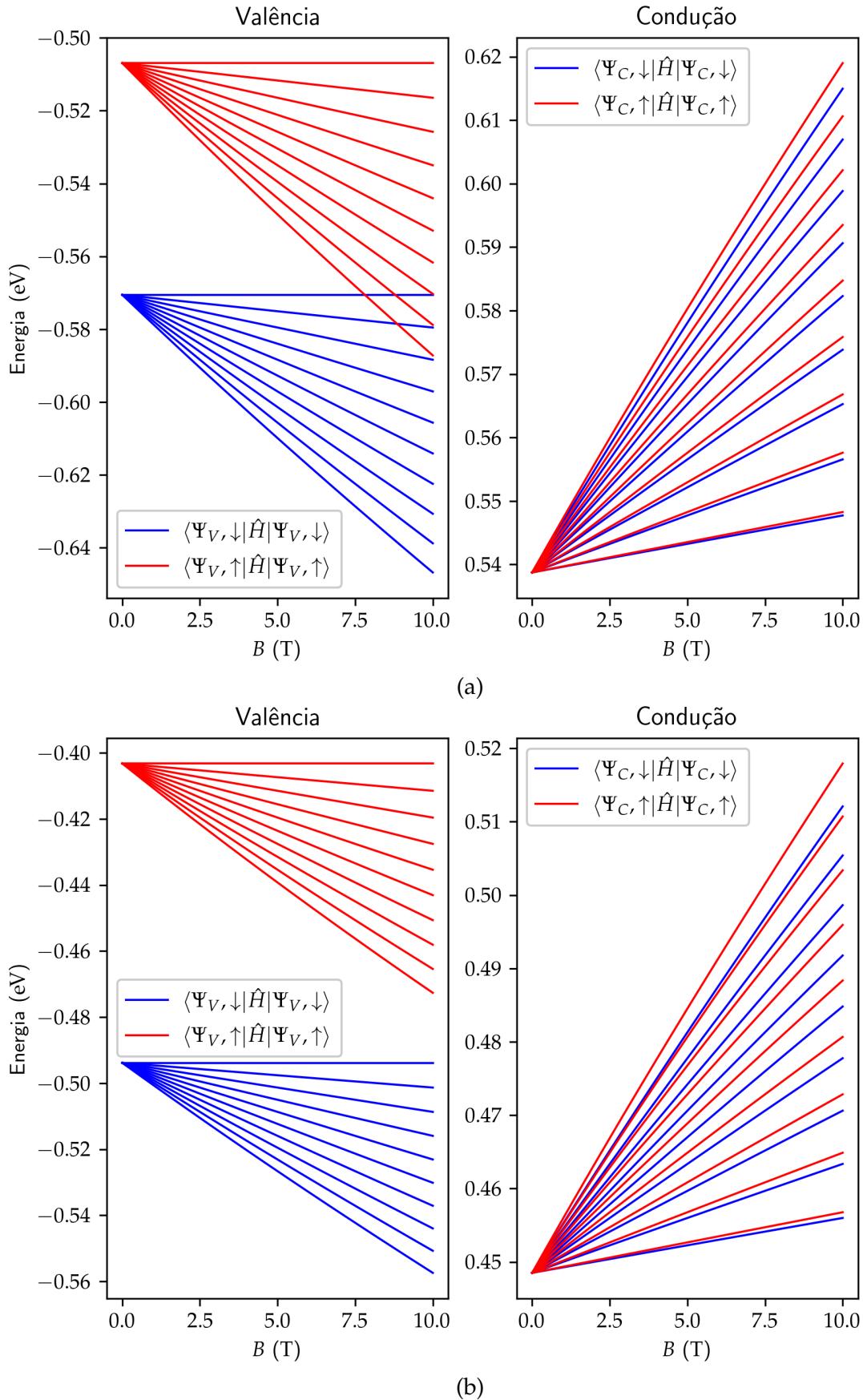


Figura 18 – Gráficos dos níveis de Landau no vale K para os materiais (a) CrS₂ e (b) CrSe₂ em termos da magnitude do campo magnético externo aplicado na direção normal a respectiva monocamada.

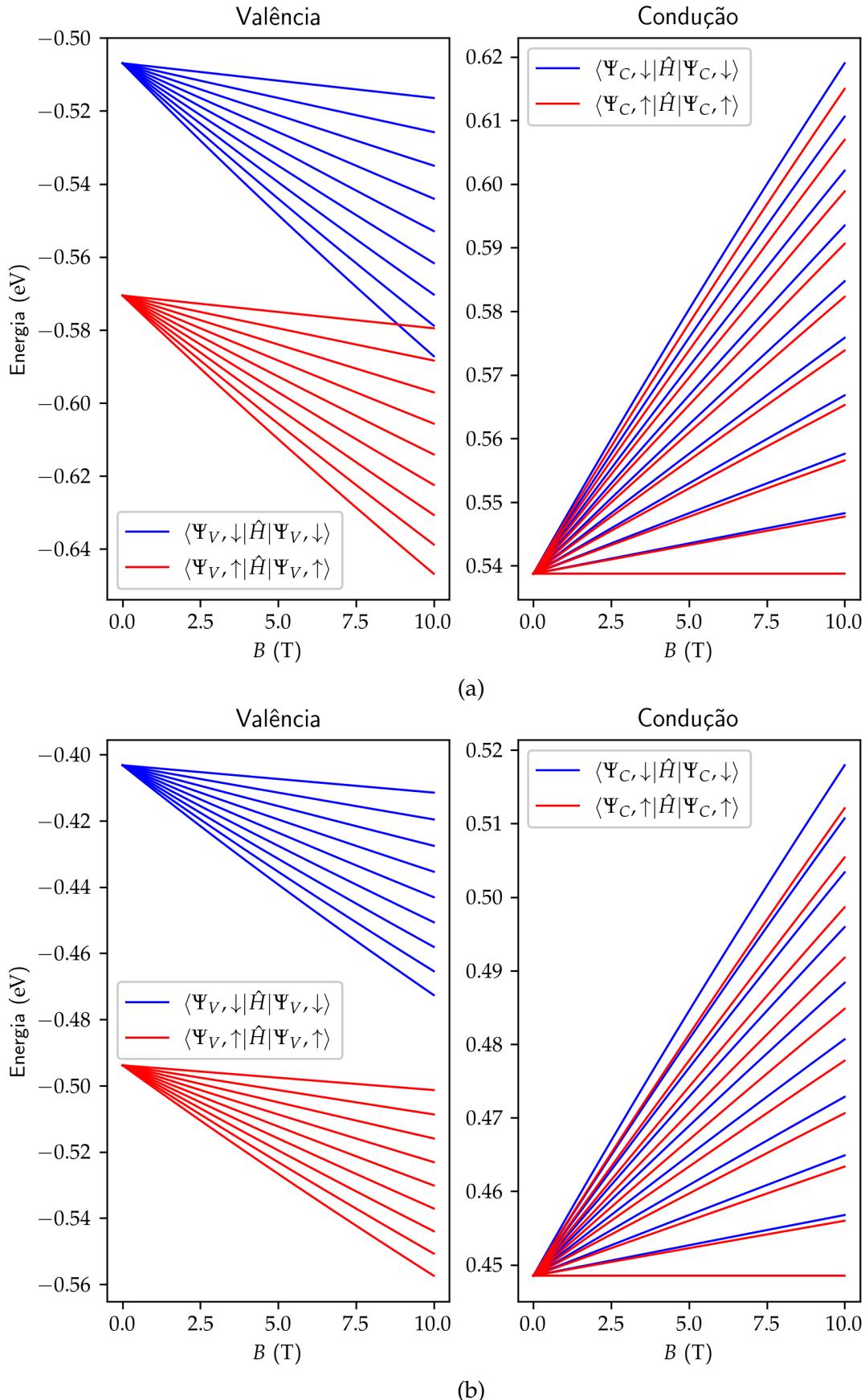


Figura 19 – Gráficos dos níveis de Landau no vale K' para os materiais (a) CrS₂ e (b) CrSe₂ em termos da magnitude do campo magnético externo aplicado na direção normal a respectiva monocamada.

3.3 Testes de Performance

Em testes de tempo de execução o programa escalou bem, sendo sua complexidade temporal $\mathcal{O}(n)$ para o tamanho de população, e $\mathcal{O}(g)$ para número de gerações decorridas. Nas figuras exibidas nesse capítulo, o tempo médio decorrido na evolução das populações de 1000 indivíduos foi de 15 segundos[†] enquanto que o tamanho da população necessário para que o processo tomasse mais de uma hora foi superior a $n = 10^5$, para o mesmo número de gerações.

Isso mostra que a implementação é escalável para grandes valores de n , o que pode ser desejável em algumas aplicações. É posto a prova também a performance da biblioteca NumPy, cujas funções e estruturas de dados implementadas se mostraram altamente otimizadas.

[†]Tempo obtido usando uma máquina com processador Intel i7-8550U com frequência máxima de 4GHz. É importante frisar aqui as evoluções das 8 populações ocorreram em 8 processos em paralelo. Assim, é possível que, para uma única população, os tempos de execução sejam ligeiramente menores.

4 Conclusão

Um algoritmo do tipo genético se mostrou eficaz na solução de problemas de otimização numérica, encontrando o máximo global com poucas iterações, sendo uma estratégia muito superior a qualquer método do tipo escalada de morro na tarefa de mapear não só os máximos locais, mas também os máximos globais.

Em problemas mais complexos, ou que demandem maior precisão, o espaço de busca pode ser restrito para uma região menor, contendo os candidatos a solução obtidos anteriormente. Na falta de candidatos, o oposto pode ser feito, expandindo a região de busca.

Uma outra característica do código implementado é a facilidade de paraleliza-lo. Em todos os resultados exibidos no capítulo anterior, a evolução de cada população foi computada de forma concorrente em 8 processos distintos, aumentando assim a amostra de indivíduos sem acréscimo significativo no tempo de execução.

Em suma, o algoritmo desenvolvido se mostrou bastante versátil, e produziu resultados favoráveis nos testes apresentados, sendo a maior dificuldade atrelada a solução do problema a determinação dos parâmetros que melhor possibilitam o sucesso na busca dos extremos. As capacidades e limitações serão postos a prova em outros testes subsequentes e nas futuras etapas do trabalho, onde será feita sua aplicação no estudo acerca de algum sistema físico.

5 Cronograma

Neste capítulo são apresentadas as atividades realizadas até o momento da confecção deste trabalho, bem como as perspectivas para o próximo período.

- Período 2021/02 — Nesse período foi feita a leitura da bibliografia citada neste texto, e então foi implementada a primeira versão do algoritmo. Posteriormente, foi feita uma série de testes com diferentes funções, dentre as quais estão as citadas nesse trabalho. Finalmente, foi confeccionado este texto.
- Período 2022/01 — No período subsequente, planeja-se escolher um problema físico que envolva a otimização de funções, realizar uma consulta sobre referências bibliográficas relacionadas, fazer a leitura dos textos encontrados e aplicar o programa na solução do problema. Por fim, o Trabalho de Conclusão de Curso será elaborado acerca dos resultados obtidos e a defesa dessa tese será apresentada.

Referências

- 1 GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. 1st. ed. USA: Addison-Wesley Longman Publishing Co., Inc., 1989. ISBN 0201157675.
- 2 HOLLAND, J. H. Genetic algorithms. *Scientific American*, JSTOR, v. 267, n. 1, p. 66–73, 1992. Disponível em: <<http://papers.cumincad.org/data/works/att/7e68.content.pdf>>.
- 3 KOLOBOV, A. V.; TOMINAGA, J. *Two-dimensional transition-metal dichalcogenides*. [S.l.]: Springer, 2016. v. 239.
- 4 LIU, G.-B. et al. Three-band tight-binding model for monolayers of group-vib transition metal dichalcogenides. *Physical Review B*, APS, v. 88, n. 8, p. 085433, 2013.
- 5 RONCARATTI, L. F.; GARGANO, R.; SILVA, G. M. e. A genetic algorithm to build diatomic potentials. *Journal of Molecular Structure: THEOCHEM*, Elsevier, v. 769, n. 1-3, p. 47–51, 2006.
- 6 DIAS, A. C. Estrutura eletrônica e propriedades magneto ópticas dos pontos quânticos de dissulfeto de molibdênio. 2016.
- 7 JR, L. A. R. et al. Predicting the equilibrium structure of organic semiconductors with genetic algorithms. *Chemical Physics Letters*, Elsevier, v. 555, p. 168–172, 2013.