



Problema de manutenção de sistema elétrico de potência Trabalho 04

Computação Evolutiva
at Universidade Federal de Uberlândia

Antonio Fernandes Valadares

15 de janeiro de 2022

Número de matrícula:
Professor:

11711ECP015
Keiji Yamanaka

Objetivo

O objetivo desse trabalho é construir um algoritmo genético para determinar quais máquinas deverão ser paradas em determinados intervalos, para que um sistema de potência possua a maior margem de segurança possível. A margem de segurança é determinada pela reserva líquida de potência Pl do sistema:

$$Pl = Pt - Pp - Pd$$

onde, Pt = potência total instalada, Pp = potência perdida por parada, Pd = potência máxima de demanda.

O sistema possui 7 máquinas com as seguintes características:

Número da máquina	Capacidade (MW)	Número de intervalos de manutenção/ano
1	20	2
2	15	2
3	35	1
4	40	1
5	15	1
6	15	1
7	10	1

Figura 1: Características das máquinas

Foram estabelecidos 4 intervalos iguais no ano para a manutenção das máquinas:

Intervalo	Demanda máxima Pd (MW)
1	80
2	90
3	65
4	70

Figura 2: Características das máquinas

A manutenção de qualquer máquina deve começar no início de um intervalo e termina no final do mesmo intervalo, ou do seguinte. A manutenção não pode ser abortada ou terminada antes do programado. A reserva líquida de cada intervalo deve ser sempre maior que 0. O melhor resultado é onde a reserva líquida de cada intervalo é a máxima.

Construção do algoritmo

O primeiro passo foi definir o formato do cromossomo de cada indivíduo da população. Defini o cromossomo de forma com que cada pedaço dele representasse uma máquina, para máquinas que necessitam de apenas um intervalo de manutenção foi simples a codificação, foram utilizados dois bits representando os quatro intervalos possíveis, dessa forma os bits 01 indicavam que a máquina estaria em manutenção no intervalo 2.

	1	2	3	4	5	6	7	cromossomo
0	000	101	10	10	00	11	01	0001011010001101
1	010	001	01	00	01	10	01	0100010100011001
2	011	100	10	10	11	00	01	0111001010110001
3	001	011	00	00	01	00	00	0010110000010000
4	101	100	10	10	11	00	10	1011001010110010
5	010	001	11	01	11	10	10	0100011101111010
6	010	011	00	10	00	01	11	0100110010000111
7	011	011	10	11	00	10	00	0110111011001000
8	010	011	11	00	01	00	00	0100111100010000
9	100	000	11	10	01	00	00	1000001110010000

Figura 3: Representação do problema como um cromossomo

Para máquinas que necessitavam de dois intervalos as combinações possíveis eram 6, dessa forma foram geradas número de 0 a 6 e convertidos em strings binárias de 3 bits.

Cálculo da aptidão

Com alguns testes realizados, notei que a soma das reservas líquidas dos intervalos sempre dava 110MW, independente da população. Dessa forma decidi utilizar como função objetivo o inverso do desvio padrão, assim quanto melhor estivesse

distribuído os 110MW entre os 4 intervalos mais apto o indivíduo seria e todos intervalos teriam uma reserva razoável.

1	2	3	4	5	6	7	cromossomo	pl1	pl2	pl3	pl4	fitness
101	000	00	11	10	10	10	1010000011101010	20	45	25	20	0.097014
101	000	10	00	11	11	11	1010001000111111	15	45	30	20	0.087287
101	010	01	10	11	11	10	1010100110111110	55	25	15	15	0.060999
001	101	11	00	00	01	01	0011011100000101	-5	35	50	30	0.049614
011	011	01	00	11	10	11	0110110100111011	30	-10	35	55	0.042400
101	010	11	11	01	01	10	1010101111010110	55	30	55	-30	0.028793
100	000	01	10	00	01	01	1000000110000101	40	-35	45	60	0.027154
100	011	11	01	00	01	01	1000111101000101	55	-40	70	25	0.023694
001	011	00	00	01	01	00	0010110000010100	-35	15	50	80	0.023368
101	101	10	10	00	10	10	1011011010001010	55	60	-50	45	0.022188

Figura 4: Cálculo do fitness para uma população de 10 indivíduos

Crossover

O crossover foi realizado utilizando elitismo e torneio. É possível configurar um parâmetro do elitismo para passar os x melhores indivíduos para a próxima população, o default é 1. Para fazer a seleção dos pais é utilizado o método do torneio, são selecionados k indivíduos aleatórios e o que possuir melhor fitness é selecionado para o cruzamento, o valor de k é um parâmetro do algoritmo e o default é 3. Com os dois pais selecionados é escolhido um valor de 0 a 1 aleatório, caso o valor seja menor que a taxa de cruzamento acontece o cruzamento e os filhos são passados adiante, caso contrário os pais são passados adiante.

Mutação

Após o cruzamento foi aplicado uma mutação na população, para cada indivíduo existe uma chance de seu cromossomo sofrer uma mutação. Caso ocorra a mutação um bit aleatório é invertido do seu cromossomo. A taxa de mutação é um parâmetro e pode ser indicado pelo usuário, o default é 0.1 que indica uma chance de mutação de 10%.

Resultados

Foram realizados varios testes, o melhor resultado encontrado foi um desvio padrão de 2.5. Dessa forma, dois intervalos iriam possuir 30MW de Pl e outros dois 25MW de Pl , forma encontradas várias respostas possíveis. Para uma decisão mais apurada seria necessário avaliar quais intervalos seria melhor ter mais potência de reserva.

```
Melhor resultado: 1010111100100110
Intervalos máquina 1: [3, 4]
Intervalos máquina 2: [2, 3]
Intervalos máquina 3: [4]
Intervalos máquina 4: [1]
Intervalos máquina 5: [3]
Intervalos máquina 6: [2]
Intervalos máquina 7: [3]
Margem de segurança dos intervalos: 30 30 25 25 Desvio padrão: 2.5
```

Figura 5: Possível resposta para o problema.

Nessa resposta temos os intervalos 1 e 2 com 30MW e os intervalos 3 e 4 com 25MW de reserva de potência.

No geral, os parâmetros utilizados foram uma taxa de crossover de 0.6, a taxa de mutação de 0.1 e O valor de k para o torneio igual a 3. Uma população de 20 indivíduos atendia bem e executando o algoritmo por 100 gerações.