Proposta de Abordagem ao Problema do Trabalho Balanceado

Gustavo Delazeri

29 de Julho de 2020

1 Formulação do Problema como Programa Inteiro

Variáveis:

- $x_{ij} \in \{0,1\}, \quad \forall i,j \mid i \in [n] \wedge j \in [m]$, onde

$$x_{ij} = \begin{cases} 1, & \text{Caso tarefa i e executada pelo operador j} \\ 0, & \text{Caso contrario} \end{cases}$$

- $w_{ijk} \in \{0,1\}, \quad \forall i,j,k \mid i,j \in [n] \wedge k \in [m] \wedge i \neq j$, onde

$$w_{ijk} = \begin{cases} 1, & \text{Caso } x_{ik} \wedge x_{jk} \text{ é verdade} \\ 0, & \text{Caso contrario} \end{cases}$$

- $y \in \mathbb{R}$, onde

$$y = \max\left(\sum_{i \in [n]} p_{ij} \cdot x_{ij}, \forall j \in [m]\right)$$

Função Objetivo:

$$\text{Min.} \quad y$$

Restrições:

$$\sum_{j \in [m]} x_{ij} = 1, \quad \forall i \in [n] \tag{1}$$

$$\sum_{i \in [n]} x_{ij} \geq 1, \quad \forall j \in [m] \tag{2}$$

$$w_{ijk} \leq (x_{ik} + x_{jk})/2, \quad \forall i,j,k \mid i,j \in [n] \wedge k \in [m] \wedge i < j \tag{3}$$

$$w_{ijk} \geq x_{ik} + x_{jk} - 1, \quad \forall i,j,k \mid i,j \in [n] \wedge k \in [m] \wedge i < j \tag{4}$$

$$w_{ijk} \leq x_{j-1k}, \quad \forall i,j,k \mid i,j \in [n] \wedge k \in [m] \wedge i < j \tag{5}$$

$$y \geq \sum_{i \in [n]} p_{ij} \cdot x_{ij}, \quad \forall j \in [m] \tag{6}$$

A restrição (1) garante que toda tarefa é executada por exatamente 1 operador. A restrição (2) garante que todo operador executa pelo menos uma tarefa. Restrições (3) e (4) formam uma conjunção: se as tarefas i e j são executadas pelo mesmo operador k, então w_{ijk} é verdade. A restrição (5) garante que operadores só executam tarefas sequenciais. Por exemplo, um operador não pode executar as tarefas 1,2 e 4. A restrição (6) define um limite inferior para a variável y, a qual representa o tempo gasto pelo operador que trabalha por mais tempo.

2 O Algoritmo Genético

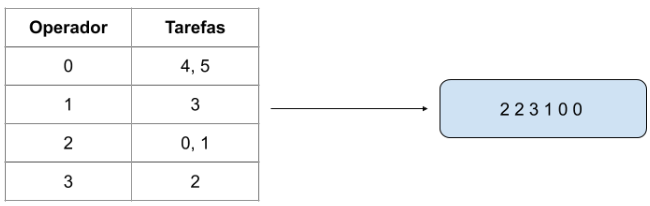
2.1 Parâmetros

A tabela abaixo apresenta os parâmetros do algoritmo e a notação adotada.

μ	Quantidade de indivíduos na população inicial
λ	Quantidade de novos individuos gerados
k	Número de participantes de um torneio aleatório
ϕ	Probabilidade de um indivíduo sofrer mutação
ω	Número máximo de gerações consecutivas que não alteram a melhor solução

2.2 Codificação de uma Solução

Uma solução para uma instância de n tarefas e m operadores é representada por um vetor v de inteiros não negativos de tamanho n. Se $v_i = j$ então o operador j executa a tarefa i. A figura abaixo ilustra a codificação de uma solução de uma instância com 6 tarefas e 4 operadores.



2.3 População Inicial

A população inicial é gerada aleatoriamente. Primeiro criam-se μ permutações e depois μ partições. Por último, associa-se a cada permutação uma partição, também de forma aleatória. A tabela abaixo ilustra o processo considerando uma instância de 8 tarefas e 4 operadores.

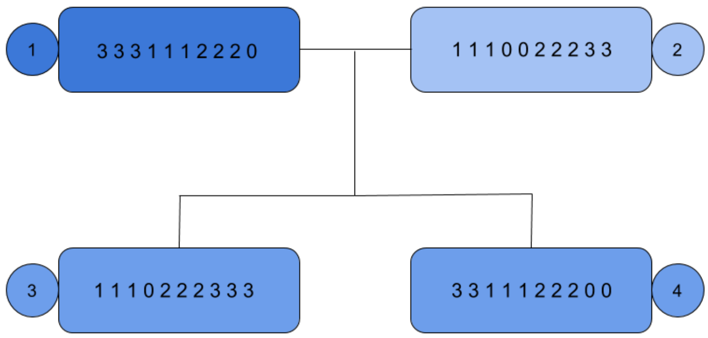
Permutação	Partição	Indivíduo Gerado
(0 3 2 1)	0 0 1 1 2 2 3 3	0 0 3 3 2 2 1 1
(2 3 0 1)	0 1 1 1 2 3 3 3	2 3 3 3 0 1 1 1
(3 1 0 2)	0 0 0 0 0 1 2 3	3 1 0 0 0 0 0 2
(1 2 3 0)	0 1 2 2 2 3 3 3	1 2 2 2 3 3 3 0

2.4 Seleção de Indivíduos para Crossover

A seleção de indivíduos para crossover implica na realização de λ k-torneios aleatórios. Um k-torneio aleatório consiste em selecionar k indivíduos da população de forma aleatória e escolher o melhor desses k indivíduos.

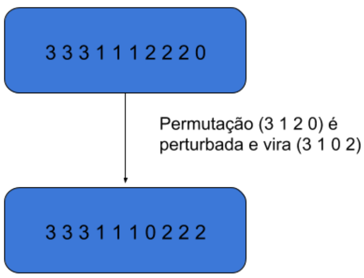
2.5 Crossover

O processo de crossover consiste em criar duas novas soluções usando a partição/permutação de um pai com a partição/permutação do outro. A figura abaixo ilustra o processo. O indivíduo 3 herda a permutação do indivíduo 2 (1 0 2 3) e a partição do indivíduo 1 (um 0, três 1's, três 2's e três 3's). Já o indivíduo 4 herda a permutação do indivíduo 1 (3 1 2 0) e a partição do indivíduo 2 (dois 0's, três 1's , três 2's e dois 3's).



2.6 Mutação

O processo de mutação consiste em aplicar uma pequena perturbação aleatória na permutação da solução. A figura abaixo ilustra o processo.



2.7 Seleção da Nova População

A seleção da nova população depende do parâmetro λ . Se λ novos indivíduos foram criados via crossover, então os λ piores indivíduos entre todos os indivíduos (geração atual e nova geração) são eliminados da população.

2.8 Critério de Parada

A execução do algoritmo para e retorna uma solução se uma ou mais das condições abaixo forem satisfeitas:

- O tempo de execução do algoritmo atingiu a marca dos 30 minutos
- ω gerações foram geradas e o valor da função objetivo não diminuiu