

Introdução

Em uma linha de produção busca-se distribuir tarefas em estações de trabalho, de modo a agilizar o processo de criação do produto final ou diminuir os custos de produção. Em centros de trabalho para deficientes (CTD), entretanto, esta associação não é simples, pois alguns trabalhadores podem executar determinadas tarefas em um tempo relativamente maior que outros ou possuem deficiências que o impeçam de realizá-las.

Com base nisso, o ALWABP-2 [1] consiste na otimização de linhas de produção com trabalhadores deficientes, em que temos que associar tarefas a trabalhadores e trabalhadores a máquinas, aumentando a contribuição dos trabalhadores deficientes mas otimizando o tempo para obtenção do produto final.

Definição do problema

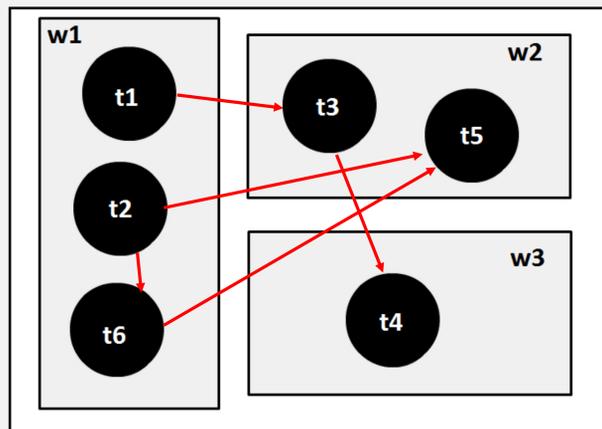


Figura 1 – Grafo representativo do problema

Seja:

- W um conjunto de trabalhadores, S um conjunto de estações de trabalho ao longo de uma esteira transportadora e T um conjunto de tarefas.
- p_{tw} o tempo da tarefa $t \in T$ executada pelo trabalhador $w \in W$.
- $G(T, E)$ um grafo em que cada aresta $(t, t') \in E \subset T \times T$ determina se uma tarefa t deve ser executada antes da tarefa t' .

Na figura 1 observa-se um exemplo de grafo representativo do problema, onde mostra-se as tarefas ($T = \{t1, t2, t3, t4, t5, t6\}$) e suas relações de dependência. Além disso, também estão representados os trabalhadores ($W = \{w1, w2, w3\}$), que abrangem as suas respectivas tarefa associadas.

Devemos distribuir as tarefas entre os trabalhadores e associar os trabalhadores às estações de trabalho, de modo que uma tarefa só seja executada em uma estação de trabalho se todas as tarefas que a precedem em $G(T, E)$ já foram executadas em estações anteriores ou na atual. Este problema é chamado de problema de balanceamento e atribuição de trabalhadores em linhas de produção (ALWABP) e como objetivamos minimizar o tempo de ciclo, devemos resolver o tipo 2 do problema (ALWABP-2).

Solução utilizada

O objetivo deste trabalho foi resolver o ALWABP-2 implementando o algoritmo iterativo genético proposto por Mutlu, Polat e Ayca [2]. O algoritmo pode ser dividido em 3 níveis como apresentados na Figura 2.

Para cada tempo de ciclo c da busca binária, o algoritmo genético é aplicado, e testa ordenações topológicas diferentes do grafo de precedências $G(T, E)$. São utilizadas operações de crossover e mutação de modo a variar a população durante o algoritmo genético.

Tendo as ordens das tarefas pré-estabelecidas, usamos a busca local para determinar em que estações e por quais trabalhadores elas serão executadas.

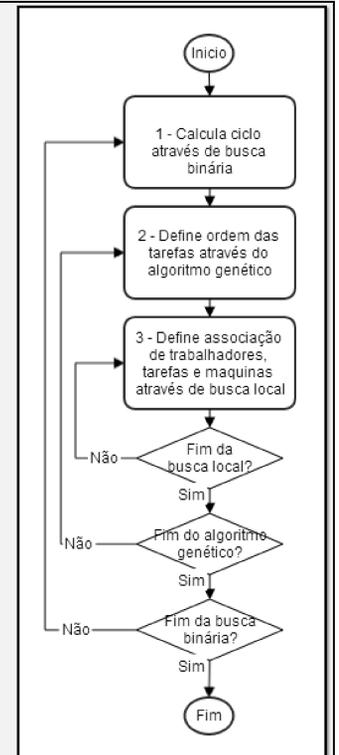


Figura 2 – Diagrama do algoritmo genético

Resultados

Foram realizados dois testes com população com $2|T|$ e $5|T|$ respectivamente, para as instancias padrões do problema usadas na literatura. Os resultados foram agrupados em grupos de 10 instâncias.

$$DR (\% \text{ de desvio}) = \frac{(ve - vo)}{vo}, \text{ onde } ve \text{ é o valor resultado e } vo \text{ é o valor ótimo.}$$

Heskia					Tonge				
Grupo	DR (low)	Tempo (s)(low)	DR (high)	Tempo (s)(high)	Grupo	DR (low)	Tempo (s)(low)	DR (high)	Tempo (s)(high)
1 – 10	0.20	0.54	0.14	2.07	1 – 10	0.37	14.67	0.34	45.06
11 – 20	0.27	0.56	0.23	1.70	11 – 20	0.53	14.73	0.45	52.49
21 – 30	0.20	0.59	0.17	2.26	21 – 30	0.40	16.19	0.35	51.37
31 – 40	0.20	0.54	0.14	2.08	31 – 40	0.44	16.06	0.39	49.44
41 – 50	0.39	0.59	0.35	1.95	41 – 50	0.64	17.32	0.55	57.20
51 – 60	0.38	0.59	0.29	2.24	51 – 60	0.66	17.60	0.64	51.85
61 – 70	0.25	0.61	0.21	1.88	61 – 70	0.59	18.79	0.54	59.90
71 – 80	0.31	0.65	0.23	2.62	71 – 80	0.63	18.39	0.56	54.66

Roszieg					Wee-Mag				
Grupo	DR (low)	Tempo (s)(low)	DR (high)	Tempo (s)(high)	Grupo	DR (low)	Tempo (s)(low)	DR (high)	Tempo (s)(high)
1 – 10	0.04	0.21	0.02	0.64	1 – 10	1.03	16.40	0.93	55.68
11 – 20	0.03	0.23	0.03	0.73	11 – 20	1.14	15.10	1.03	52.45
21 – 30	0.03	0.24	0.02	0.79	21 – 30	1.11	19.95	1.02	65.31
31 – 40	0.03	0.21	0.02	0.73	31 – 40	1.11	19.07	1.06	65.95
41 – 50	0.08	0.21	0.09	0.70	41 – 50	1.31	15.22	1.30	46.67
51 – 60	0.13	0.24	0.08	0.85	51 – 60	1.42	17.03	1.33	51.81
61 – 70	0.06	0.24	0.04	0.85	61 – 70	1.56	20.23	1.41	78.58
71 – 80	0.06	0.22	0.04	0.64	71 – 80	1.58	17.40	1.44	62.79

Conclusões

Conclui-se com os resultados obtidos que o algoritmo genético encontra resultados razoáveis para as instâncias do problema e possui uma grande melhora quando se aumenta o tamanho da população de cromossomos.

Em relação a implementação do artigo do Mutlu [2], observa-se que os resultados deles ainda são melhores, o que pode acontecer devido a dificuldade de identificar determinados detalhes obscuros do algoritmo. Outro ponto interessante para melhoria da nossa implementação é a questão do tempo de execução que ainda está alto para grandes populações.

Referências:

- [1] MIRALLES, Cristobal et al. Advantages of assembly lines in sheltered work centres for disabled. A case study. International Journal of Production Economics, v. 110, n. 1, p. 187-197, 2007.
[2] MUTLU, Özcan; POLAT, Olcay; AYCA SUPCILLER, Aliye. An iterative genetic algorithm for the assembly line worker assignment and balancing problem of type-II. Computers & Operations Research, 2012.