

10^o

FEPEG FÓRUM

ENSINO • PESQUISA
EXTENSÃO • GESTÃO
RESPONSABILIDADE SOCIAL: INDISSOCIABILIDADE
ENSINO, PESQUISA E EXTENSÃO UNIVERSITÁRIA



ISSN 1806-549 X

Autor(es): RAMÓN SOUZA SILVA RODRIGUES, VICTOR DE FREITAS ARRUDA, MATEUS FELLIPE ALVES LOPES, DIEGO LEAL MAIA, JOÃO BATISTA MENDES

Investigação e Proposta de resolução do *Job Shop Scheduling Problem* pelo Algoritmo de Evolução Diferencial

Introdução

Este trabalho tem como objetivo apresentar os estudos da Iniciação Científica Voluntária sobre a viabilidade de resolução do *Job Shop Scheduling Problem* (JSP), ou Problema de Alocação, através de métodos de algoritmos evolutivos, em especial o algoritmo de Evolução Diferencial (ED), que se baseia no mecanismo de busca do operador de mutação diferencial. Este operador gera novas soluções a partir de vetores diferenciais construídos com pares de soluções candidatas retiradas da própria população. Por ter características tais como, robustez, fácil implementação, versatilidade e eficiência o ED mostra-se como sendo um algoritmo capaz de gerar bons resultados para o JSP, assim como abordado por Tasgetiren (2006).

De acordo com Dasgupta o problema pode ser definido como:

Existe um número de agentes e um número de tarefas. Podemos alocar qualquer agente a qualquer uma das tarefas, porém, cada alocação tem um custo que pode variar dependendo da tarefa e agente específicos. É necessário que todas as tarefas sejam feitas, designando exatamente um agente para cada tarefa de modo que o custo total (a soma) de todas as alocações seja minimizado. (Dasgupta et al., 2008).

O *Job Shop* clássico (JSP) pode ser apresentado da seguinte maneira: Existem n jobs, cada um composto de várias operações que determinam o tempo de execução de cada *job*. Esses *jobs* devem ser processados em m máquinas sendo que cada operação utiliza uma das m máquinas de acordo com o tempo de duração desse processo. Cada máquina pode processar no máximo uma tarefa por vez, como ilustrado na fig. 1. O problema consiste em encontrar um escalonamento das operações que minimize o custo somado de todas as alocações, respeitando as restrições existentes.

O Problema de Alocação é encontrado com frequência em aplicações práticas em várias áreas do conhecimento humano, como em indústrias (planejamento da produção) ou nas instituições de ensino (problema de alocação de salas de aula). Porém, sua característica é de um problema que se enquadra na categoria dos problemas *NP-Difíceis*, ou seja, é difícil encontrar uma solução ótima para o mesmo em tempo hábil (que varia conforme o tamanho do problema) em problemas de larga escala. Um estudo detalhado sobre o problema de alocação pode ser visto em Soares (2011).

Material e métodos

Foi realizada uma pesquisa bibliográfica acerca do problema de alocação de recursos, identificando os principais trabalhos encontrados na literatura especializada. Posteriormente, foi feita uma modelagem matemática do problema, onde foram identificados a função objetivo e as principais restrições encontradas na literatura. No passo seguinte, foi estudado um algoritmo evolutivo para resolução do problema modelado anteriormente.

Resultados e discussão

Através dos estudos e pesquisas sobre o problema JSP foi escolhida uma abordagem entre várias para solução do problema através de Algoritmos Genéticos (AGs). Estes possuem mecanismos de busca que evitam soluções consideradas ótimas locais, ou seja, melhor que soluções semelhantes. Baseados na Teoria da Evolução de Darwin, os AGs tratam de problemas complexos como o JSP a partir de uma estrutura que compreende cada possível solução como um indivíduo, com suas características específicas, da população de uma geração que evolui a cada geração nova gerada através da combinação dos melhores indivíduos da geração anterior, aproximando-se assim de uma solução ótima de maneira eficiente e com um baixo custo computacional do que um método matemático que conclui com o encontro da solução ótima.

Estudou-se o algoritmo evolutivo de Evolução Diferencial. Ele é um algoritmo de otimização simples e eficiente que tem sido aplicado com sucesso em diversos problemas de otimização. A primeira publicação sobre esse algoritmo ocorreu em 1995, em um relatório técnico de Rainer Storn e Kenneth Price (Storn and Price, 1995), criadores do algoritmo. Embora o ED seja classificado como um algoritmo evolutivo, a mutação diferencial não tem base ou inspiração em nenhum processo natural. A forma como esse operador promove mutações nos indivíduos se sustenta em argumentos matemáticos e heurísticos que mostram sua adequabilidade para a otimização de funções e não exatamente em argumentos derivados de processos da natureza. Apesar disso o ED segue uma linha histórica de algoritmos e



métodos que evoluem uma população de soluções. Por isso, ele é classificado como uma instância dos algoritmos evolutivos.

O ED funciona da seguinte forma: Seu mecanismo de busca utiliza vetores diferenciais criados a partir de vetores da própria população. Dois indivíduos são aleatoriamente selecionados da população corrente, originando um vetor com a diferença entre estes dois vetores. Este vetor, por sua vez, é somado a um terceiro indivíduo, também escolhido aleatoriamente, produzindo então uma nova solução mutante através de uma perturbação da diferença entre dois indivíduos. Esse processo pode ser ilustrado pela seguinte formulação:

$$V^{(q+1)} = X_{\alpha}^{(q)} + F(X_{\beta}^{(q)} - X_{\gamma}^{(q)})$$

Onde X_{α} , X_{β} , X_{γ} são os vetores escolhidos aleatoriamente e distintos entre si. Em que a diferença é multiplicada por um $F > 0$ da geração q , sendo denotada por diferença ponderada. Este processo resulta o vetor doador $V^{(q+1)}$.

Esta formulação é empregada para geração de uma solução mutante que é recombinada com os indivíduos da população corrente, produzindo a descendência ou população teste. Por fim cada solução é comparada com seu correspondente na população corrente. Se a solução teste é melhor (mais apta) do que seu correspondente, a solução corrente é removida e substituída pela solução teste. Caso contrário, a solução teste é descartada. O processo se repete até que algum critério de parada estabelecido seja satisfeito. O funcionamento do ED é apresentado a seguir:

Pseudocódigo do algoritmo ED:

- 1) **Inicializar**, randomicamente, a população inicial;
- 2) **Avaliar** cada indivíduo;
- 3) **Repetir** até que o critério de parada seja satisfeito;
 - a) **Para cada** indivíduo
 - **Selecionar** *randomicamente 3 indivíduos* da população;
 - **Aplicar** diferenciação, mutação;
 - **Comparar** indivíduo com a sua *versão experimental* e selecionar o de menor custo;
 - **Avaliar** o custo do indivíduo selecionado;
 - b) **Fim Para**
- 4) **Fim Repetir**

Até o presente momento foram feitas atividades de estudo do problema de alocação de recursos, acerca do algoritmo de evolução diferencial e sobre a representação a ser adotada para desenvolver uma adaptação do ED para o problema de alocação de recursos. Também foram pesquisados e baixados diversos *benchmarks* acerca de problemas de alocação de recursos existentes na literatura para utilização na implementação do algoritmo. Além disso, foi implementada uma rotina para computar o *makespan* total de uma solução para o problema. Esta rotina será utilizada pelo algoritmo ED para avaliar a qualidade de uma solução para o problema.

A próxima etapa do presente trabalho consiste na implementação do operador de evolução diferencial gerando como esperado novas soluções para o JSP. Em seguida o algoritmo será aperfeiçoado com introdução de técnicas de busca local objetivando aprimorar a qualidade das soluções geradas.

Considerações finais

Através das análises e estudos das características do algoritmo de evolução diferencial, tais como, robustez, fácil implementação, versatilidade e eficiência, observou-se que este algoritmo pode ser empregado na resolução de problemas como o JSP.

O algoritmo ED possui várias qualidades desejáveis em algoritmos de otimização de propósito geral:

1. Capacidade de adaptar-se à estrutura da função;
2. Poucos parâmetros de controle a serem ajustados, uma vez que os tamanhos de passo da mutação são auto adaptados;
3. Simplicidade de implementação.



Agradecimentos

O autor gostaria de agradecer a todos os seus professores do seu curso superior e a Universidade Estadual de Montes Claros pela base de conhecimento na área para que esse estudo pudesse ser desenvolvido.

Referências bibliográficas

DASGUPTA, D. et al. **A Comparison of Multiobjective Evolutionary Algorithms with Informed Initialization and Kuhn-Munkres Algorithm for The Sailor Assignment Problem.** In: PROCEEDINGS OF THE 2008 GECCO CONFERENCE COMPANION ON GENETIC AND EVOLUTIONARY COMPUTATION, 2008, New York, NY, USA: ACM, 2008, p. 2129–2134.

SOARES, H. C. A. **Um estudo sobre o Problema de Alocação.** 2011. Monografia (Ciência da Computação), Universidade Federal de São Paulo, São Paulo.

STORN, R. M; PRICE, K. V. **Differential evolution** - a simple and efficient adaptive scheme for global optimization over continuous spaces, Technical Report TR-95-012, International Computer Science Institute (ICSI), University of California, Berkeley, 1995.

TASGETIREN, M. F. et al. **A Particle Swarm Optimization and Differential Evolution Algorithms for Job Shop Scheduling Problem.** vol. 3, n. 2, p. 120-135, 2006.

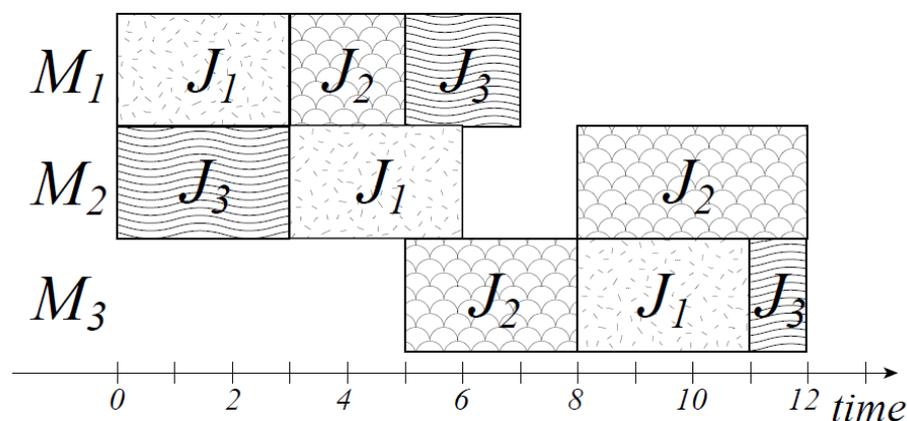


Figura 1. Escalonamento de máquinas e tarefas