



# Aplikácia evolučných algoritmov a hlbokého učenia pre inteligentné plánovanie výroby

**Prípadová štúdia:**

"Research on the Application of Evolutionary Algorithms Enhanced by Deep Learning in Intelligent Manufacturing Scheduling"

**Autori článku:** Daqi Sun, Haijuan Zang

**Autori prezentácie:** Bc. Lukáš Vangor, Bc. Filip Kaža



# Úvod a kontext

Cieľ článku:

- Minimalizovať celkový čas dokončenia (makespan)
- Znížiť prestoje strojov
- Zvýšiť efektivitu výroby

Prečo evolučné algoritmy?

- Efektívne riešenie komplexných optimalizačných problémov
- Schopnosť globálneho prehľadávania riešení



# Evolučné algoritmy v článku

Evolučné algoritmy (EA):

- Používanie operácií ako crossover a mutácia
- Problém: Zasekávajú sa v lokálnych optimách

Ďalšie metódy:

- Particle Swarm Optimization, Artificial Bee Colony Algorithm, Whale Optimization Algorithm, Gray Wolf Algorithm



# Integrácia hlbokého učenia

Actor-Critic model:

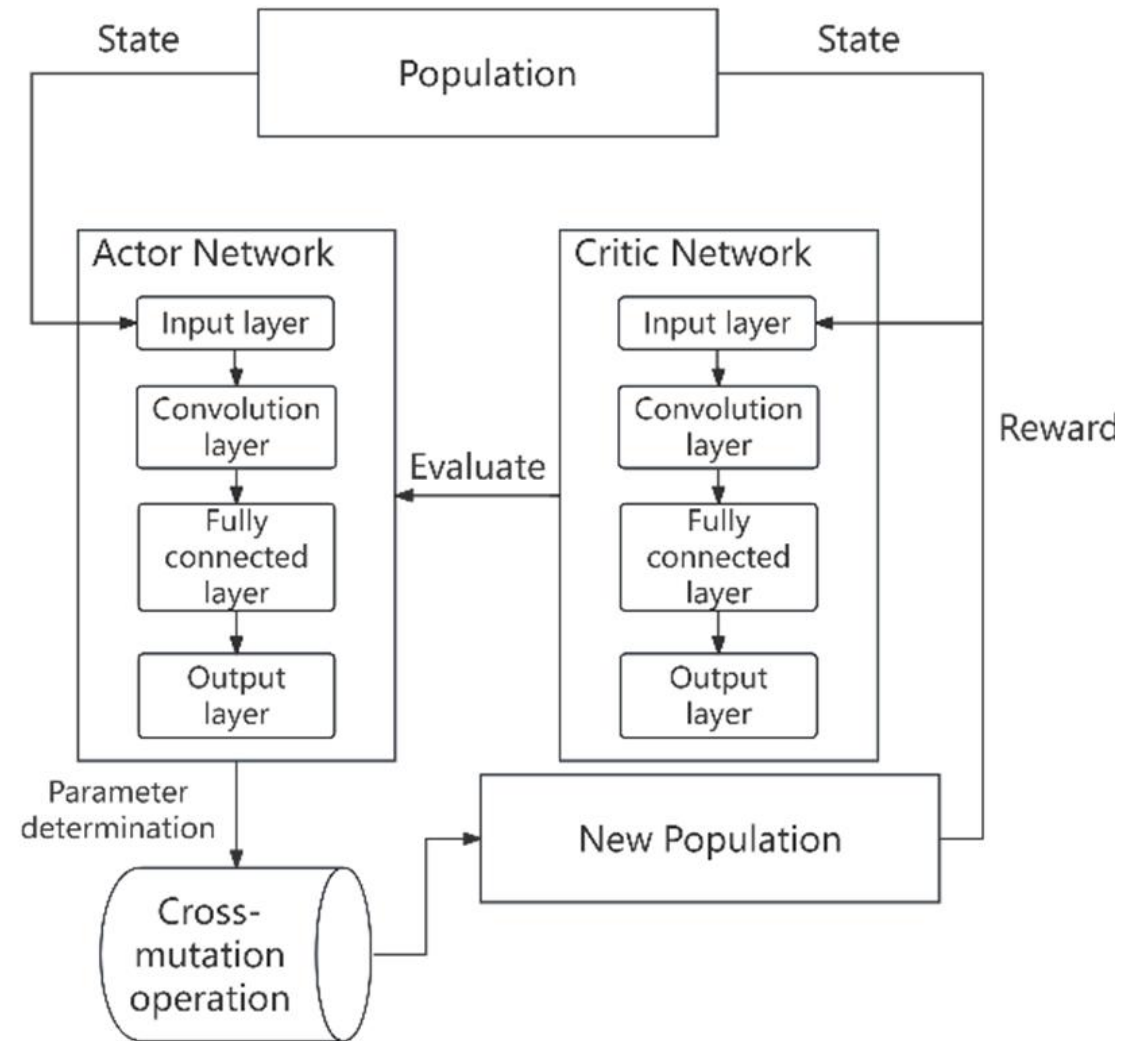
- Actor sieť: Generuje akcie (upravuje crossover a mutačné pravdepodobnosti)
- Critic sieť: Hodnotí vykonané akcie a poskytuje spätnú väzbu

Výhody:

- Adaptívna zmena parametrov
- Rýchlejšia konvergencia
- Zabránenie uviaznutiu v lokálnych optimách



Obr. 1 – Architektúra Actor-Critic modelu: Actor sieť upravuje parametre (pravdepodobnosť kríženia a mutácie), Critic ich hodnotí na základe získanej odmeny.



Obr. 1



# Genetické operácie a kódovanie

- Sekvenčné kódovanie (určuje poradie operácií)
- Kódovanie priradenia strojov

Crossover:

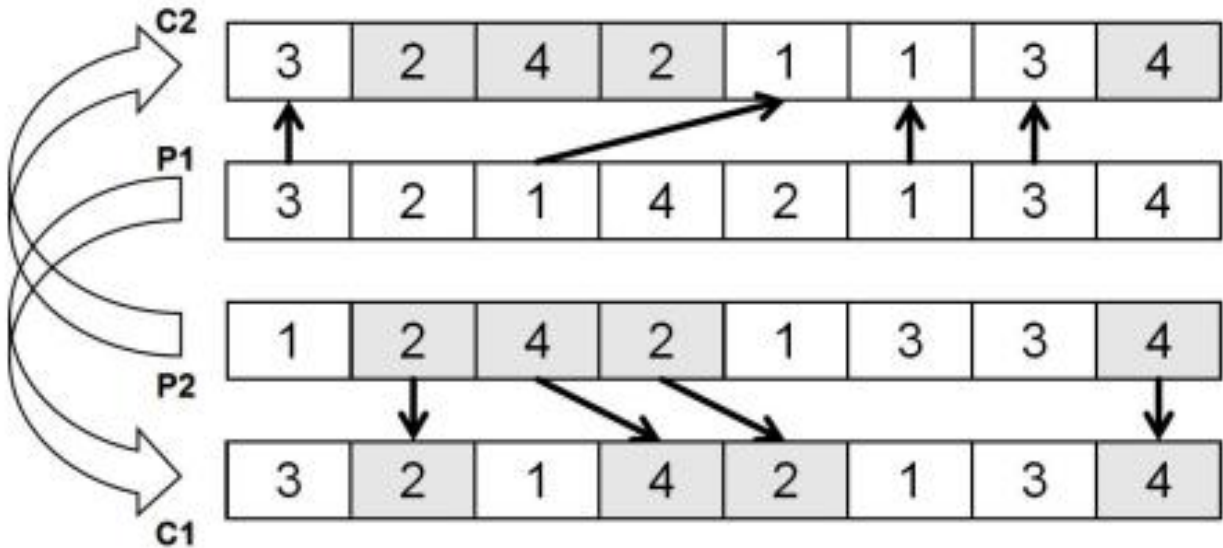
- Partial Order Crossover (POX) a Uniform Crossover

Mutácie:

- Reverzná mutácia
- Mutácia na základe minimálneho času potrebného na spracovanie



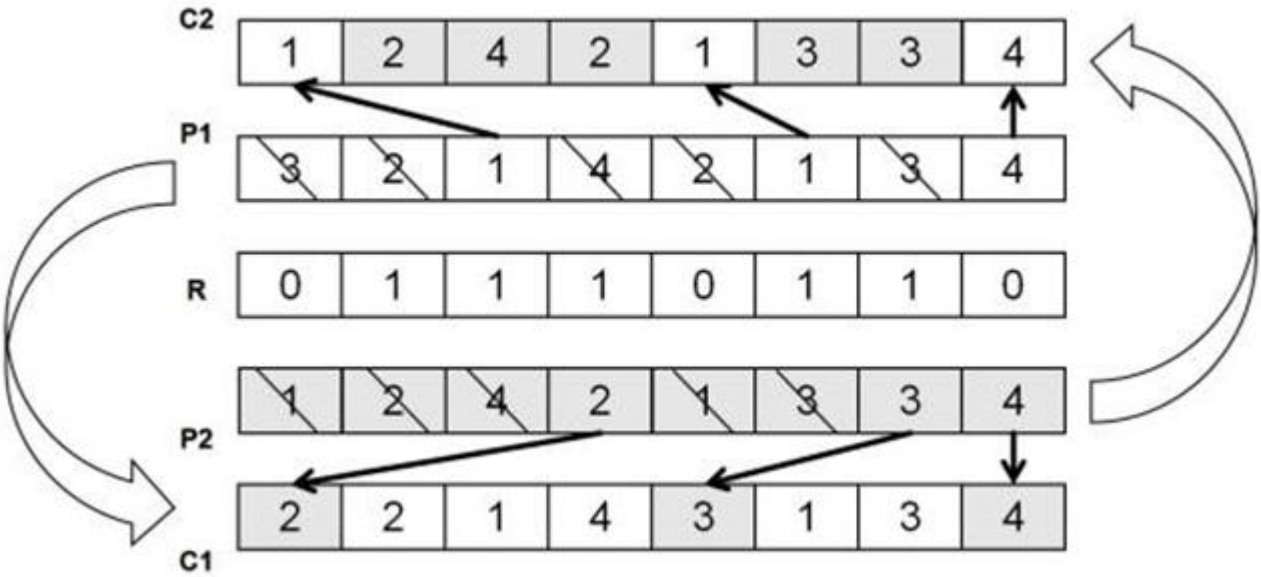
Obr. 2 – POX crossover: znázorňuje spôsob, akým sa vyberajú a kombinujú časti chromozómov počas kríženia operácií



Obr. 2



Obr. 3 – Uniform crossover:  
rovnorné náhodné rozdelenie  
génov medzi rodičovské chromozómy



Obr. 3





# Paralelné spracovanie pomocou K-means

Cieľ: Zvýšiť diverzitu populácie a zlepšiť globálne prehľadávanie riešení

Proces:

1. Inicializácia populácie
2. Aplikácia K-means klastrovania na rozdelenie do subpopulácií
3. Paralelné vykonávanie genetických operácií v jednotlivých skupinách

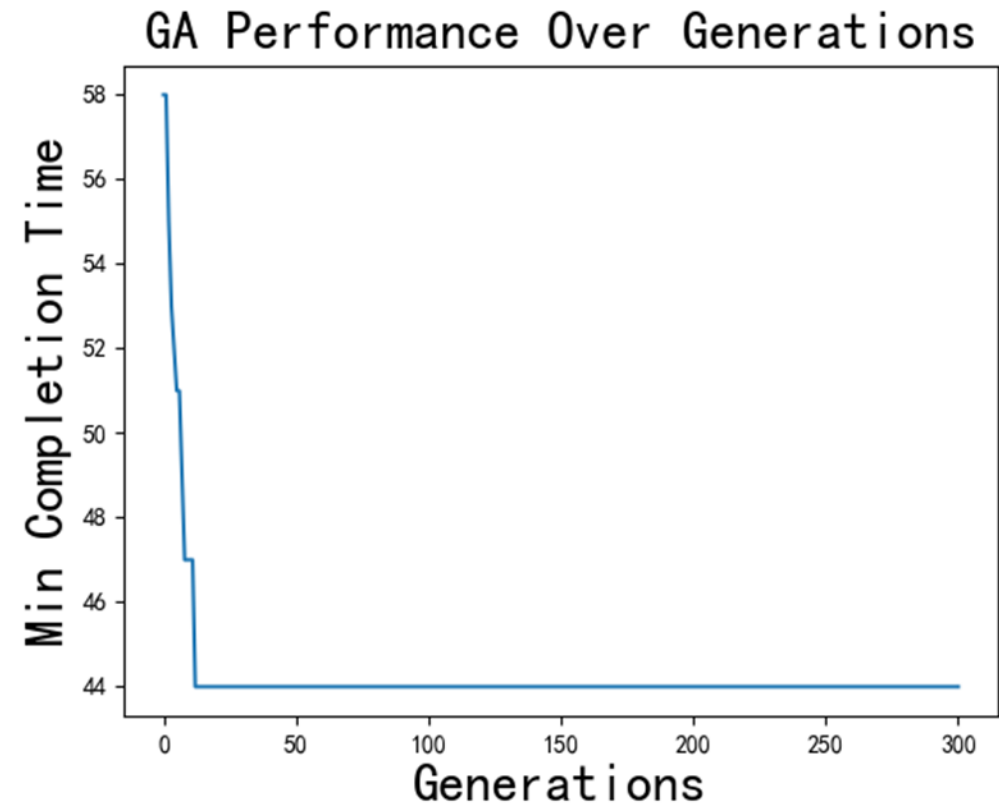
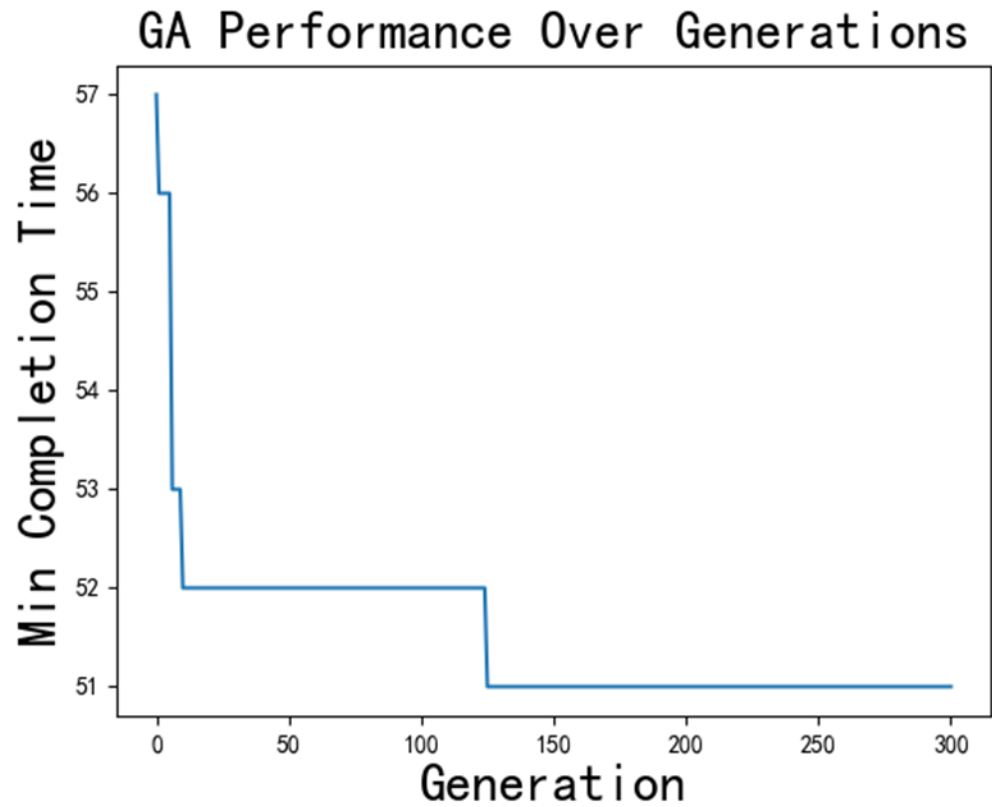
# Experimentálne výsledky a analýza

Porovnanie s tradičným EA:

- Tradičný EA: Pomalý pokles makespanu, lokálne optimá
- AC-FJSP (s Actor-Critic): Rýchly pokles makespanu, efektívnejšie využitie zdrojov

Vizualizácie:

- Iteračné krivky a Ganttové diagramy ukazujúce zníženie prestojov strojov a kompaktné plánovanie



Klasický genetický  
algoritmus vs AC-FJSP  
algoritmus

- Klasický - postupná optimalizácia s uviaznutím v lokálnom minime
- AC-FJSP - rýchla konvergencia k lepšiemu globálnemu riešeniu

# Záver a budúce smery

## Prínosy:

- Dynamická adaptácia parametrov genetického algoritmu
- Zlepšenie globálneho prehľadávania riešení
- Efektívne rozdelenie výpočtovej záťaže pomocou K-means

## Budúce zlepšenia:

- Jednoduchšie štruktúry a modulárny dizajn
- Využitie distribuovaného výpočtu a hardvérového zrýchlenia



Ďakujeme za pozornost



# Odkaz na článok

<https://ieeexplore.ieee.org/abstract/document/10754379>