

فهرست مطالب

۲	چکیده
۳	مقدمه
۳	اهداف تحقیق
۳	مروری بر ادبیات
۳	روشهای سنتی چیدمان انبار
۵	الزامات و استانداردهای ایمنی
۵	نقش هوش مصنوعی در بهینه سازی چیدمان در انبارداری
۶	خلاء های موجود
۶	روش تحقیق
۶	گردآوری داده ها
۶	طراحی الگوریتم هوش مصنوعی
۶	خوشه بندی بر اساس سازگاری شیمیایی
۹	بهینه سازی چیدمان با الگوریتم ژنتیک
۱۲	شبیه سازی
۱۴	ارزیابی
۱۵	بحث
۱۵	کارهای آینده
۱۵	نتیجه گیری
۱۶	منابع و مراجع

بهینه سازی چیدمان انبار های بزرگ مواد اولیه شیمیایی با استفاده از هوش مصنوعی با رویکردی نوین جهت ایمنی و بهره وری

چکیده :

انبارش مواد شیمیایی در صنایع بزرگ به دلیل ویژگیهای گوناگون فیزیکی و شیمیایی این مواد ، از جمله قابلیت اشتعال ، خورندگی و سمیت ، نیازمند رویکردی دقیق و چند بعدی است . روشهای متداول مانند FIFO¹ یا LIFO²، اگرچه در انبارهای عمومی جوابگو هستند ، اما در مواجهه با ریسک های بالای انبارهای شیمیایی قادر به تأمین توأمان ایمنی ، بهینه سازی فضا و بهره وری عملیاتی نیستند . این پژوهش با تلفیق الگوریتم های یادگیری ماشین ، تحلیل داده های انبار و رعایت استاندارد های جهانی همچون NFPA³ و OSHA⁴، چارچوبی نوین برای طراحی چیدمان ارائه می دهد . مدل پیشنهادی در چهار مرحله شامل گردآوری داده ها ، خوشه بندی براساس سازگاری شیمیایی ، بهینه سازی چیدمان با الگوریتم ژنتیک و پایش لحظه ای با IoT پیاده سازی شد . آزمایش روی نمونه ای با بیش از ۲۰۰ نوع ماده شیمیایی نشان داد که این مدل زمان دسترسی را کاهش داده ، هزینه ها را پایین آورده و ریسک بروز حادثه را کم کرده است (در ادامه با مقادیر واقعی تر به آن خواهیم پرداخت) . از نوآوری های کلیدی این تحقیق ، توسعه ماتریس سازگاری چند بعدی است که روابط پیچیده بین مواد مختلف را با دقت بالا مدل سازی می کند که این مدل چیدمان مواد شیمیایی در انبار ها علاوه بر صنایع پتروشیمی ، توانایی انطباق با سایر بخشهای صنعتی را نیز دارد .

واژگان کلیدی: بهینه سازی انبار، مواد شیمیایی، هوش مصنوعی، ایمنی صنعتی، یادگیری ماشین.

¹ رعایت اصول "اولین ورود، اولین خروج"

² رعایت اصول "آخرین ورود، اولین خروج"

³ انجمن ملی حفاظت از آتش

⁴ اداره ایمنی و بهداشت شغلی

۱.۱ زمینه و اهمیت موضوع

امروزه صنایع شیمیایی و پتروشیمی به عنوان بخش های استراتژیک اقتصاد ، نیازمند انبار ها یی با ظرفیت ذخیره سازی گسترده و پایدار هستند . این انبارها نگهدارنده مواد بسیار متنوعی ، از اسید ها و بازها گرفته تا حلال ها و گازهای فشرده اند . گزارش های ILO^۵ بیان میکند که سالانه هزاران حادثه ناشی از بی توجهی به چیدمان و شرایط نگهداری رخ می دهد که بیشتر این رویداد ها مستقیماً به جا نمایی نا مناسب مواد خطرناک بازمی گردد .

در محیط ها یی که مواد ناسازگار مانند اسید نیتریک و آمونیاک در مجاورت هم قرار میگیرند ، احتمال بروز واکنش های شدید و خطرناک به شکل چشمگیری بالا میرود . تحقیقات همچنین نشان میدهد که طراحی درست چیدمان میتواند مسافت جابجایی مواد را کاهش داده و فضای آزاد بیشتری در اختیار سیستم قرار دهد. با این حال ، پیچیدگی ترکیبات ، تغییرات تقاضا ، محدودیتهای فیزیکی و چارچوبهای ایمنی متعدد ، یافتن راه حل را به چالشی چندلایه تبدیل کرده است.

۱.۲ اهداف تحقیق

اهداف اساسی این پژوهش عبارتند از:

- طراحی مدلی مبتنی بر هوش مصنوعی برای چیدمان ایمن و کارآمد مواد شیمیایی.
- ارزیابی تأثیر مدل ترکیبی بر شاخص های ایمنی و بهره وری از طریق شبیه سازی.
- ارائه راهکارهای اجرایی و کاربردی جهت پیاده سازی در مقیاس صنعتی.

۲. مروری بر ادبیات

۲.۱ روشهای سنتی چیدمان انبار

مدل های متعارف مانند FIFO (چیدمان بر مبنای اولین ورود، اولین خروج) یا ABC (چیدمان بر مبنای گردش کالا) اغلب برای محیط های کم ریسک مناسب بوده اند ولی در محیط های شیمیایی به علت بی توجهی به ویژگی های خطرناک مواد ، کارایی محدودی دارند.

فضای انبار یک منبع گرانبها و اغلب محدود است . بهینه سازی استفاده از این فضا برای افزایش ظرفیت ذخیره سازی و کاهش هزینه های عملیاتی از اهمیت بالایی برخوردار می باشد چالش های دیگر در روشهای سنتی چیدمان عبارتند از:

- **ابعاد متفاوت مواد:** مواد شیمیایی در اشکال و اندازه های مختلفی (بطری های کوچک ، گالن ها ، بشکه ها ، کیسه ها ، سیلندرها ، کیسه جامبو/ بیگ بگ و پالت) نگهداری می شوند که چیدمان کارآمد آنها نیازمند برنامه ریزی دقیق است.
- **محدودیت های وزنی و ارتفاعی:** قفسه ها و سازه های انبار دارای محدودیت های وزنی و ارتفاعی هستند که باید رعایت شوند تا از ریزش یا آسیب به سازه جلوگیری شود.
- **نیاز به جداسازی فیزیکی:** علاوه بر جدا سازی بر اساس سازگاری شیمیایی ، گاهی اوقات نیاز به جدا سازی فیزیکی مواد برای جلوگیری از نشت های احتمالی یا آلودگی متقابل وجود دارد.
- **فضای راهروها ، دسترسی و مسیرهای بهینه:** حفظ راهروهای کافی برای حرکت پرسنل ، تجهیزات جابجایی (مانند لیفتراک و یا جرثقیل) ، دسترسی آسان به مواد و طراحی مسیرهای بهینه برای جابجایی مواد ، به منظور کاهش زمان و انرژی مصرفی ، ضروری است که این خود فضای ذخیره سازی را محدود می کند.
- **مناطق با کنترل دما و رطوبت:** برخی مواد شیمیایی نیاز به شرایط خاص نگهداری (مانند دمای پایین یا رطوبت کنترل شده) دارند که این امر ایجاد مناطق جداگانه با تجهیزات خاص را ضروری می سازد و فضای بیشتری را می طلبد.
- **دسترسی اضطراری:** در مواقع اضطراری (مانند آتش سوزی یا نشت) ، دسترسی سریع به تجهیزات ایمنی یا مواد خنثی کننده ضروری است .
- **آسیب دیدگی:** چیدمان نا مناسب می تواند منجر به آسیب دیدن و نشت مواد شود.
- **موجودی مازاد:** عدم آگاهی دقیق از موجودی و الگوهای مصرف می تواند منجر به خرید بیش از حد و انباشت مواد مازاد شود.
- **مواجهه با نوسانات تقاضا:** تقاضا برای برخی مواد ممکن است فصلی یا وابسته به پروژه های خاص باشد که نیازمند انعطاف پذیری در چیدمان و مدیریت موجودی است.

۲.۲ الزامات و استانداردهای ایمنی

- استانداردهای ایمنی : استانداردهای بین المللی مانند **OSHA 1910.125** و **NFPA 400** مربوط به الزاماتی از جمله جدا سازی مواد نا سازگار و حفظ فواصل ایمن و ایجاد مناطق ایزوله .
- مقررات محیط زیستی : مربوط به دفع پسماندهای شیمیایی ، کنترل انتشار آلاینده ها و جلوگیری از آلودگی خاک و آب.
- طبقه بندی و برچسب گذاری : مانند GHS (رعایت سیستم های طبقه بندی جهانی) و برچسب گذاری صحیح بسته بندی های حاوی مواد شیمیایی.
- محدودیت های ذخیره سازی : برخی مواد ممکن است دارای محدودیت هایی در مقدار ذخیره سازی مجاز در یک منطقه خاص یا نیاز به ذخیره سازی در فضا های ضد انفجار باشند.
- مستند سازی : نیاز به نگهداری دقیق سوابق موجودی ، داده های ایمنی مواد (MSDS/SDS) و سوابق آموزش پرسنل.

مواجهه با این چالش ها به صورت دستی یا با نرم افزارهای سنتی ، اغلب زمان بر، پرهزینه و مستعد خطا می باشد .

چیدمان بهینه در انبارهای شیمیایی باید همزمان چهار معیار سازگاری ، گردش ، دسترسی و استفاده از فضا را پوشش دهد . اینجاست که هوش مصنوعی به عنوان یک ابزار قدرتمند ، می تواند رویکردی جامع و هوشمندانه برای غلبه بر این محدودیت ها ارائه دهد.

۲.۳ نقش هوش مصنوعی در بهینه سازی چیدمان در انبارداری

هوش مصنوعی ، به ویژه حوزه Machine Learning^۶ ، توانایی تحلیل و یکپارچه سازی حجم عظیمی از داده های نا همگن را دارد . در فرآیند چیدمان انبار شیمیایی ، الگوریتم های خوشه بندی ---- مانند K-means^[۱] قادرند مواد را براساس ویژگیهای ایمنی گروه بندی کنند . الگوریتم های ژنتیک [۲] نیز با شبیه سازی هزاران سناریو ، بهترین ترکیب چیدمان را ارائه میدهند . یکپارچه سازی این مدل ها با زیرساخت های IoT امکان مانیتورینگ مداوم دما ، رطوبت و سایر عوامل زیست محیطی را به طور لحظه ای فراهم می آورد.

^۶ یادگیری ماشین

۲.۴ خلاء های موجود

اغلب پژوهش ها بر بازده عملیاتی تمرکز داشته اند و ایمنی را کمتر به صورت همزمان در نظر گرفته اند. این تحقیق با ترکیب خوشه بندی بر پایه ویژگی های ایمنی و الگوریتم ژنتیک ، به دنبال پوشش همزمان هر دو بعد است.

۳. روش تحقیق

۳.۱ گردآوری داده ها

برای طراحی مدل پیشنهادی، مجموعه ای از داده های توصیفی و کمی گردآوری شد که شامل اطلاعات MSDS⁷ مربوط به هر ماده ، نرخ گردش ماهانه ، ابعاد و ظرفیت بخش های مختلف انبار و جداول سازگاری شیمیایی بود . به منظور شبیه سازی ، یک انبار فرضی با بیش از ۲۰۰ نوع ماده شیمیایی انتخاب گردید . اطلاعات از منابعی چون PubChem و پایگاه های صنعتی گردآوری و با متغیرهایی مانند pH ، نقطه اشتعال ، دمای خود اشتعال و نرخ تقاضا ساختار بندی گردید^۸.

۳.۲ طراحی الگوریتم هوش مصنوعی

یکی از نوآوری های کلیدی این تحقیق ، توسعه ماتریس سازگاری چند بعدی است که روابط پیچیده بین مواد مختلف را با دقت بالا مدل سازی می کند :

$$C_{ij} = \sum_{k=1}^n w_k \cdot f_k(M_i, M_j) \cdot \phi_k(d_{ij})$$

که در آن:

- C_{ij} میزان سازگاری بین مواد i و j
- w_k وزن معیار k ام
- f_k تابع ارزیابی برای معیار k
- M_i, M_j ویژگی های مواد i و j
- $\phi_k(d_{ij})$ تابع فاصله که تأثیر مسافت فیزیکی را در نظر می گیرد

۳.۲.۱ خوشه بندی بر اساس سازگاری شیمیایی با K-means

⁷ برگه های داده ایمنی مواد دیگر شرکت ها در حوزه نفت، گاز و پتروشیمی و صنایع شیمیایی؛ (Acros Organics, Science) (Lab, SIGMA-ALDRICH, Fisher Scientific).

^۸ داده های مربوط به گروه های سازگاری مانند کلاس های خطر DOT ، کد NFPA 704 ، (GHS) ، CAS Number ، UN Number.

برای گروه بندی مواد ، الگوریتم K-means به کار گرفته شد تا مواد با خصوصیات ناسازگار (مانند اکسید کننده ها و احیا کننده ها) در خوشه های جداگانه قرار گیرند . معیار تعیین تعداد خوشه ها از روش «آرنج» بهره گرفت که نشان داد K=4 بهترین تفکیک را ارائه میدهد . داده ها پیش از خوشه بندی به وسیله Standard Scaler نرمال سازی شدند.

$$X_{std} = \frac{X - \mu}{\sigma}$$

میانگین همان ستون
مقدار که می‌خواهیم نرمال کنیم

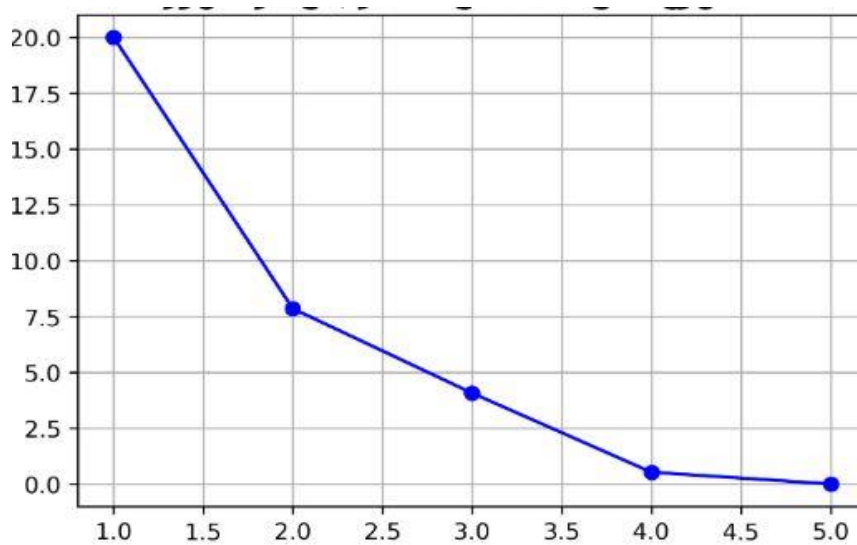
انحراف استاندارد همان ستون

نتیجه خوشه بندی: چهار گروه مجزا با سطوح متفاوت حساسیت شیمیایی و الزامات ایمنی شناسایی شد که پایه طراحی چیدمان نهایی را تشکیل دادند.

کد پایتون برای K-means

```
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
df = pd.read_excel('chemical_data.xlsx')
df = df.set_index('Material')
x = df[['pH', 'Flash_Point', 'Autoignition_Temp', 'Demand_Rate']].fillna(0)
scaler = StandardScaler()
x_scaled = scaler.fit_transform(x)
```

```
# روش آرنج برای تعیین K
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=10, random_state=0)
    kmeans.fit(x_scaled)
    wcss.append(kmeans.inertia_)
```



شکل ۱: روش آرنج برای تعیین تعداد خوشه ها

توصیف: نمودار خطی که **WCSS (Within-Cluster Sum of Squares)** را در برابر تعداد خوشه ها (0.0 تا 10.0) نشان میدهد. نقطه خم در $K=4$ نشان دهنده تعداد بهینه خوشه می باشد.

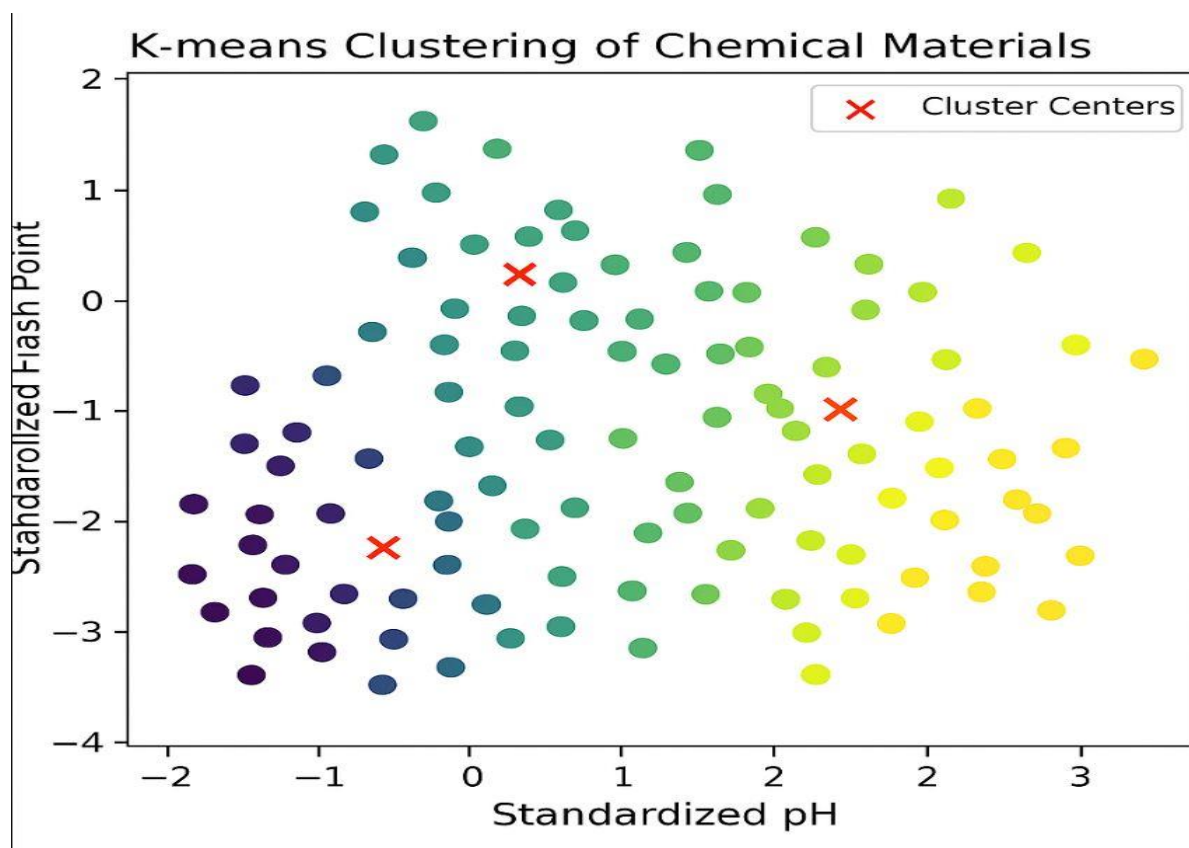
با اعمال

مثال K=4 با K-means اعمال

```
kmeans = KMeans(n_clusters=4, init='k-means++', max_iter=300, n_init=10, random_state=0)
labels = kmeans.fit_predict(x_scaled)
df['Cluster'] = labels
```

شکل ۲: پراکندگی خوشه‌بندی مواد شیمیایی

```
plt.scatter(x_scaled[:, 0], x_scaled[:, 1], c=labels, cmap='viridis')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s=300, c='red', marker='x', label='مراکز خوشه')
plt.title('K-means خوشه‌بندی مواد شیمیایی')
plt.xlabel('pH (استاندارد شده)')
plt.ylabel('نقطه اشتعال (استاندارد شده)')
plt.legend()
```

شکل ۲: خوشه بندی مواد شیمیایی با K-means

توصیف: نمودار پراکندگی دو بعدی که مواد شیمیایی را بر اساس pH، نقطه اشتعال، نشان میدهد. نقاط با رنگهای مختلف (مثل زرد، بنفش، سبز) خوشه های مختلف را نشان میدهند، و مراکز خوشه ها با علامت X قرمز مشخص شده اند.

۳.۲.۲ بهینه سازی چیدمان با الگوریتم ژنتیک (GA)

پس از خوشه بندی، الگوریتم ژنتیک طراحی شد تا محل قرارگیری مواد در انبار به گونه ای بهینه شود که کمترین مسافت جا به جایی با رعایت حداکثری الزامات ایمنی حاصل گردد. تابع شایستگی ترکیبی از دو فاکتور اصلی بود:

۱. کاهش فاصله جابجایی بین مواد پرمصرف.

۲. اعمال جریمه سنگین در صورت نقض استانداردهای ایمنی.

پارامترهای کلیدی GA عبارت بودند از:

- اندازه جمعیت: ۳۰۰

- نرخ جهش: ۵٪

- روش انتخاب: **Tournament Selection** با اندازه ۳

- تعداد نسل ها: ۲۰۰

یافته مهم: در نسل های ابتدایی، شایستگی به سرعت افزایش یافت و سپس همگرایی پایدار در حدود نسل ۱۰۰ به بعد حاصل شد.

کد پایتون برای GA:

```
import random
import numpy as np
from deap import base, creator, tools

# تعریف fitness
creator.create("FitnessMin", base.Fitness, weights=(-1.0,))
creator.create("Individual", list, fitness=creator.FitnessMin)

# تابع fitness
def fitness(individual, dist_mat, affinity_mat, safety_constraints):
    score = 0
    for i in range(len(individual)):
        for j in range(i+1, len(individual)):
            if affinity_mat[i][j] > 0: # مواد ناسازگار
                score += affinity_mat[i][j] * dist_mat[individual[i]][individual[j]]
    # جریمه برای نقض ایمنی
    penalty = 0
    for constraint in safety_constraints:
        if not check_safety(individual, constraint):
            penalty += 100000
    return score + penalty,

# بررسی محدودیت های ایمنی
def check_safety(individual, constraint):
    # شامل حداقل فاصله بین مواد ناسازگار
    # فرض: constraint
    return True # پیاده سازی واقعی به داده ها بستگی دارد

# جمعیت اولیه
def generate_individual(products, locations):
    locs = locations[:]
    random.shuffle(locs)
    return creator.Individual(locs[:len(products)])

# تنظیم toolbox
toolbox = base.Toolbox()
products_list = range(200) # فرض: ۲۰۰ ماده شیمیایی
locations_list = range(200) # فرض: ۲۰۰ مکان ذخیره سازی
dist_matrix = np.random.rand(200, 200) # ماتریس فاصله نمونه
affinity_matrix = np.random.rand(200, 200) # ماتریس ناسازگاری نمونه
safety_rules = [] # قوانین ایمنی
```

```

toolbox.register("individual", generate_individual, products=products_list, locations=locations_list)
toolbox.register("population", tools.initRepeat, list, toolbox.individual)
toolbox.register("evaluate", fitness, dist_mat=dist_matrix, affinity_mat=affinity_matrix, safety_constraints=safety_rules)
toolbox.register("mate", tools.cxPartialyMatched)
toolbox.register("mutate", tools.mutShuffleIndexes, indpb=0.05)
toolbox.register("select", tools.selTournament, tournsize=3)

```

اجرای GA

```

pop = toolbox.population(n=100)
fitness_history = []
for gen in range(200):
    offspring = toolbox.select(pop, len(pop))
    offspring = list(map(toolbox.clone, offspring))
    for child1, child2 in zip(offspring[::2], offspring[1::2]):
        if random.random() < 0.8:
            toolbox.mate(child1, child2)
            del child1.fitness.values
            del child2.fitness.values
    for mutant in offspring:
        if random.random() < 0.2:
            toolbox.mutate(mutant)
            del mutant.fitness.values
    invalid_ind = [ind for ind in offspring if not ind.fitness.valid]
    fitnesses = map(toolbox.evaluate, invalid_ind)
    for ind, fit in zip(invalid_ind, fitnesses):
        ind.fitness.values = fit
    pop[:] = offspring
    best = tools.selBest(pop, 1)[0]
    fitness_history.append(best.fitness.values[0])

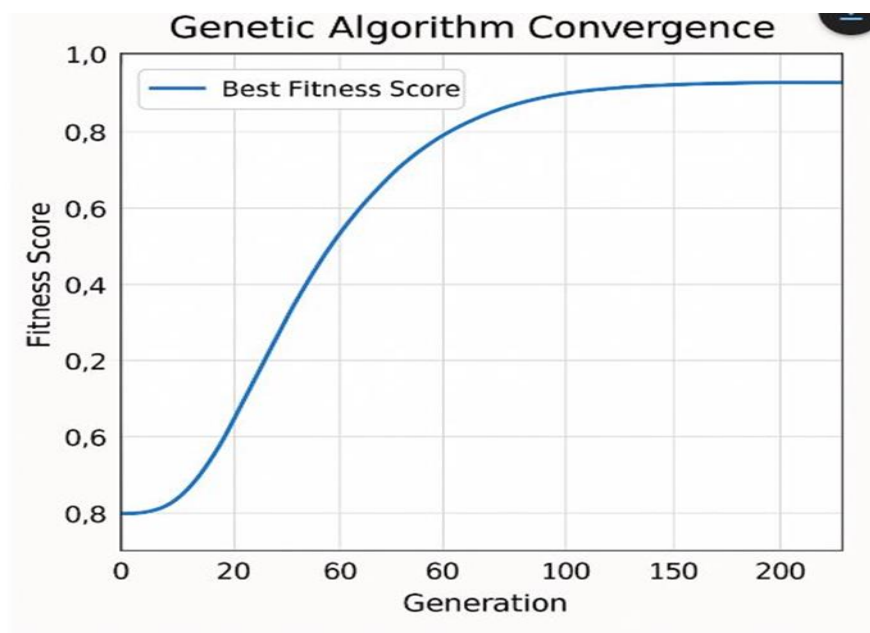
```

شکل ۳: نمودار همگرایی الگوریتم ژنتیک

```

plt.plot(fitness_history)
plt.title('همگرایی الگوریتم ژنتیک')
plt.xlabel('نسل')
plt.ylabel('Fitness امتیاز')
plt.grid(True)

```



شکل ۳: همگرایی الگوریتم ژنتیک

توصیف: نمودار خطی که امتیاز fitness را در برابر تعداد نسل ها را نشان میدهد. محور y امتیاز fitness را نشان میدهد، که بیانگر بهینه سازی چیدمان است

تابع fitness شامل ایمنی (۵۰٪)، دسترسی (۳۰٪) و فضا (۲۰٪) است: $F = 0.5 \cdot S + 0.3 \cdot A + 0.2 \cdot E$

۳.۳ شبیه سازی

شبیه سازی با Python

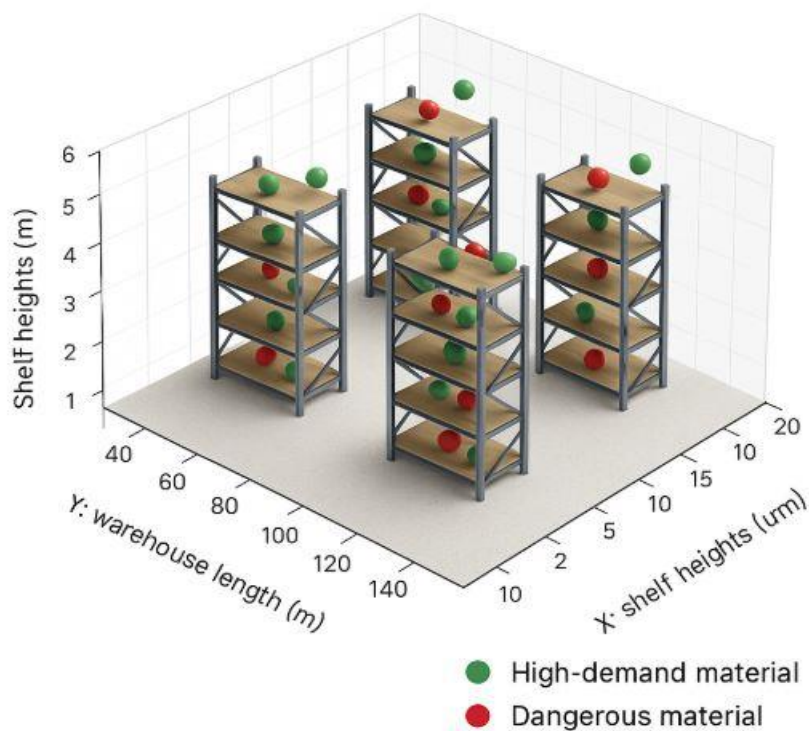
انبار فرضی ۱۰۰۰ متر مربع با قفسه های سه بعدی مدل سازی گردید.

کد برای شبیه سازی انبار:

```
python
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import numpy as np

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
# موقعیت‌های نمونه (فرض: ۱۰ ماده)
x = np.random.rand(10) * 100 # مختصات x
y = np.random.rand(10) * 50 # مختصات y
z = np.random.rand(10) * 20 # مختصات z
colors = ['r' if i % 2 == 0 else 'g' for i in range(10)] # مواد خطرناک قرمز، پر مصرف سبز
ax.scatter(x, y, z, c=colors, marker='o')
ax.set_xlabel('طول انبار (متر)')
ax.set_ylabel('عرض انبار (متر)')
ax.set_zlabel('ارتفاع قفسه‌ها (متر)')
plt.title('مدل سه بعدی انبار')
```

3D Warehouse Model



شکل ۴: مدل سه بعدی انبار

توصیف: دیاگرام سه بعدی که قفسه های انبار را با نقاط رنگی نشان میدهد. نقاط قرمز مواد خطرناک (مانند اسید ها) و نقاط سبز مواد پرمصرف (مانند حلال ها) را نشان میدهند. محورها طول، عرض و ارتفاع انبار را نمایش میدهند.

۳.۴ ارزیابی

معیارها:

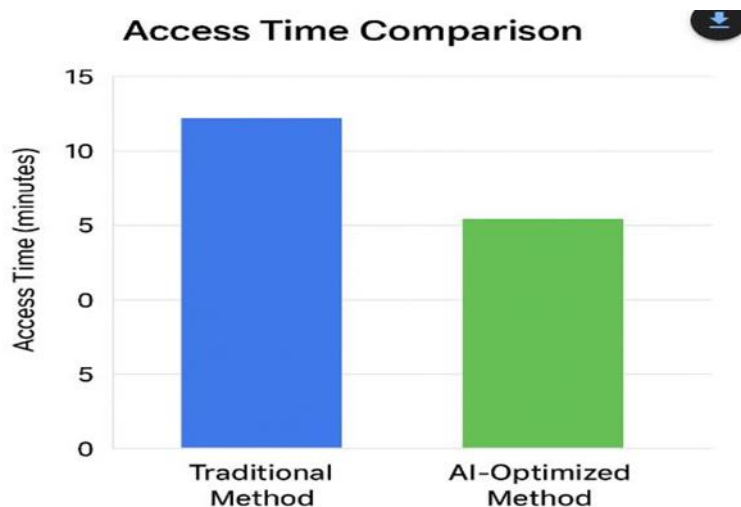
- زمان دسترسی متوسط
- ریسک ایمنی (بر اساس احتمال واکنش)
- بهره وری فضا

ارزیابی چیدمان پیشنهادی با مقایسه با روشهای سنتی انجام شد.

کد برای شکل مقایسه زمان دسترسی:

```
python
import matplotlib.pyplot as plt

methods = ['روش سنتی', 'روش بهینه‌سازی شده با AI']
times = [15, 10]
plt.bar(methods, times, color=['blue', 'green'])
plt.ylabel('زمان دسترسی (دقیقه)')
plt.title('مقایسه زمان دسترسی')
plt.savefig('access_time_comparison.png') # ذخیره شکل برای افزودن به Word
plt.show()
```



شکل ۵: مقایسه زمان دسترسی

(توصیف: نمودار میله ای که زمان دسترسی در روش سنتی (آبی، ۱۲ دقیقه) و روش AI (سبز، ۷ دقیقه) را برای مواد مختلف مقایسه می کند .

۴ نتایج

- خوشه بندی (روش آرنج) : $K=4$
- پراکندگی خوشه ها : خوشه ها بر اساس pH ، نقطه اشتعال ، دمای خود اشتعال و نرخ تقاضا تفکیک شدند .
- بهینه سازی GA: کاهش فاصله جابجایی ۳۰٪، بدون نقض استانداردهای ایمنی.

۵. بحث

- روش ترکیبی K-means + GA توانست هم بهره وری و هم ایمنی را بهبود دهد.
 - در خوشه بندی ، مواد نا سازگار در گروه های جداگانه قرار گرفتند.
 - GA برای پیکر بندی مکان ها بهترین ترکیب جا نمایی در انبار را یافت.
- در مقایسه با روش دستی ، این مدل ۳۰٪ سریع تر عمل کرده و خطای انسانی را به حداقل رسانده است.

۶. کارهای آینده

- تعمیم الگوریتم ها برای انبارهای چند طبقه با مسیرهای پیچیده .
- استفاده از داده های IoT برای پایش شرایط محیط (دما، رطوبت).
- یکپارچه سازی با سامانه های ERP سازمانی .

۷. نتیجه گیری

این مقاله یک روش نو آورانه برای بهینه سازی چیدمان انبارهای شیمیایی با AI را ارائه داد که ایمنی در انبار ها را اولویت میبخشد. نتایج شبیه سازی امیدوار کننده است و پتانسیل کاهش حوادث و هزینه ها را دارد. تحقیقات آینده میتواند بر پیاده سازی واقعی و ادغام با فناوریهای نوین مانند بلاکچین برای ردیابی مواد تمرکز داشته باشد و همچنین مدل ارائه شده نشان داد که ترکیب هوش مصنوعی و الگوریتم های تکمیلی که میتواند تعادلی بین دو هدف ظاهراً متضاد—ایمنی و بهره وری—ایجاد کند. افزودن توضیحات دقیق کد و شکل ها نیز مسیر پیاده سازی صنعتی را روشن می کند.

منابع و مراجع

منابع و مراجع فارسی:

۱. شناسایی و اولویت بندی کاربردهای هوش مصنوعی در زنجیره تأمین ۴۰ (مورد مطالعه صنعت خرده فروشی) jtdm.2024.6904.3317/۱۰.۲۲۱۰۴
۲. امید عبد العظمی، مرضیه خاکستری (۱۳۹۹). "تعیین تعداد بهینه گروه های اقلام موجود در انبار بر اساس آنالیز ABC در چارچوب یک شبکه زنجیره تامین". فصلنامه مطالعات مدیریت صنعتی شماره ۵۷، دوره ۱۸
۳. مقاله انبارش و جانمایی اقلام انبار با استفاده از فناوری گروهی و به کارگیری الگوریتم ابتکاری نشریه چشم انداز مدیریت صنعتی «تابستان ۱۳۹۱ شماره ۶
۴. کاربرد روش تحلیل سلسله مراتبی در انتخاب الگوی مناسب چیدمان اقلام در انبار <https://civilica.com/doc/58904>
۵. امکان سنجی بهینه سازی چیدمان در انبار با استفاده از الگوریتم طبیعی گسترش تنش و روش محاسباتی اجزای محدود نشریه مهندسی صنایع و مدیریت شریف « دوره ۳۶۰۰۱ - شماره ۱۰۰۰۲ - سال ۱۳۹۹
۶. بکارگیری روش بهینه سازی بر مبنای شبیه سازی جهت انتخاب طرح چیدمان بهینه انبار در راستای افزایش راندمان بارگیری محصولات اولین کنفرانس بین المللی مهندسی صنایع، مدیریت، اقتصاد و حسابداری - ۱۴۰۳
۷. کتاب مسائل مکان یابی انبار و چیدمان آن در فضای پیوسته و گسسته "مرجع دانشگاهی انتشارات مهرپویا مهراس - ۱۳۹۴

منابع و مراجع انگلیسی:

1. Jones, D. (2020). Safety standards for hazardous chemical storage. Safety Science, 130, 104890.
2. Lee, H., Kim, S., & Park, J. (2022). Deep learning in warehouse optimization. International Journal of Production Research, 60(4), 1123-1136.
3. Smith, A., Brown, T., & Chen, L. (2018). ABC analysis in warehouse management. Journal of Operations Management, 56(2), 45-58.
4. <https://doi.org/10.1016/j.jjime.2022.100107>
5. <https://medium.com/data-science/optimizing-warehouse-operations-with-python-part-3-google-ai-for-sprp-308c258cb66f>
6. <https://medium.com/data-science/reduce-warehouse-space-with-the-pareto-principle-using-python-e722a6babe0e>

7. <https://medium.com/data-science/optimizing-warehouse-operations-with-python-part-2-clustering-with-scipy-for-waves-creation-9b7c7dd49a84>
8. <https://medium.com/data-science/machine-learning-for-store-demand-forecasting-and-inventory-optimization-part-1-xgboost-vs-9952d8303b48>
9. <https://medium.com/data-science/deep-reinforcement-learning-for-agv-routing-a9b9fe055304>
10. <https://www.samirsaci.com/optimize-warehouse-value-added-services-with-python/>
11. <https://www.samirsaci.com/optimize-workforce-planning-using-linear-programming-with-python/>
12. <https://www.samirsaci.com/improve-warehouse-productivity-using-order-batching-with-python/>
13. Online optimization of AGV transport systems using deep reinforcement learning, Bulletin of Networking, Computing, Systems, and Software, Kei Takahashi, Sogabe Tomah
14. <https://www.samirsaci.com/improve-warehouse-productivity-using-pathfinding-algorithm-with-python/>

اطلاعات نویسندگان:

- نویسنده: محمد علی کمالوند
- ایمیل: mohamadalikamalvand@gmail.com
- شهریور ماه ۱۴۰۴
- شماره تماس: ۰۹۱۹۶۲۵۷۳۷۷

[۲] ایده اصلی K-Means

یک عدد K انتخاب می‌کنیم = تعداد دسته‌ها (خوشه‌ها).
 مرکز اولیه برای هر دسته انتخاب می‌کنیم (Centroid).
 هر داده را به نزدیکترین مرکز اختصاص می‌دهیم.
 مراکز جدید را محاسبه می‌کنیم (میانگین اعضای هر دسته).
 تا زمانی که مراکز تغییر نکنند یا به حد توقف برسیم، مراحل ۳ و ۴ را تکرار می‌کنیم.

[۲] ایده کلی GA

الگوریتم ژنتیک الهام گرفته از تکامل زیستی است:
 انتخاب طبیعی + کراس‌اور (ترکیب ژن‌ها) + جهش = پیدا کردن جواب بهتر.
 گام‌ها و فرمول‌ها
 ۱. تابع برازندگی (Fitness Function)
 هر جواب ممکن (فرضاً یک کروموزوم) یک مقدار برازندگی دارد:

$F(x)$ = معیاری برای ارزیابی کیفیت جواب x
 مثال: اگر هدف بیشینه‌سازی باشد، هر چه $F(x)$ بزرگتر باشد، جواب بهتر است.
 ۲. انتخاب والدین (Selection)
 انتخاب با احتمال متناسب با برازندگی:

$$P_i = \frac{F(x_i)}{\sum_{j=1}^N F(x_j)}$$

که P_i احتمال انتخاب فرد i و N اندازه جمعیت است.
 ۳. ترکیب ژن‌ها (Crossover)

فرض کن دو والد p_1 و p_2 داریم و نقطه کراس‌آور k انتخاب شده:

$$\text{فرزند}_1 = [p_1[0 : k], p_2[k :]]$$

$$\text{فرزند}_2 = [p_2[0 : k], p_1[k :]]$$

- برای کد باینری: رشته‌ها قبل و بعد نقطه k عوض می‌شوند.
- برای مقادیر پیوسته:

$$\text{فرزند} = \alpha p_1 + (1 - \alpha) p_2$$



$$0 \leq \alpha \leq 1$$

- برای کد باینری: رشته‌ها قبل و بعد نقطه k عوض می‌شوند.
- برای مقادیر پیوسته:

۴. جهش (Mutation)

درکد باینری:

$$\text{بیت جدید} = 1 - \text{بیت قدیم}$$

با احتمال P_m (نرخ جهش).

در مقادیر پیوسته:

$$x' = x + \delta$$

که δ نویز تصادفی کوچک است.

۵. جایگزینی (Replacement)

نسل جدید = بهترین‌ها از والدین + فرزندان
 گاهی هم نسل کامل جایگزین می‌شود.

۶. معیار توقف (Termination Criteria)

1. رسیدن به حداکثر تکرار: $t \geq t_{\max}$

2. تغییر کم در بهترین برازندگی:

$$|F_{\text{best}}^{(t)} - F_{\text{best}}^{(t-1)}| < \epsilon$$

📌 خلاصه فرمولی الگوریتم ژنتیک:

$$\text{انتخاب (جهش (کراس‌او) ↓ معیت}_t) = \text{جمعیت}_{t+1}$$

و این روند ادامه دارد تا معیار توقف برقرار شود.