

## ДОДАТОК А

Графічний матеріал атестаційної роботи

ГЮИК.504300.014

(Позначення документу)

Харківський національний університет радіоелектроніки

«ЗАТВЕРДЖУЮ»  
Керівник атестаційної роботи  
проф. Безкоровайний В.В.

МЕТОДИ СТРУКТУРНО-ПАРАМЕТРИЧНОЇ ОПТИМІЗАЦІЇ ДИСКРЕТНИХ  
ВИРОБНИЧИХ ТЕХНОЛОГІЧНИХ ПРОЦЕСІВ

Графічний матеріал

ЛИСТ ЗАТВЕРДЖЕННЯ

ГЮИК.504300.014-ЛЗ

РОЗРОБИВ:  
ст. гр. СПРМ-18-2  
Сідорчук Є. І.

2020 р.

ЗАТВЕРДЖЕНО  
ГЮИК.504300.014-ЛЗ

МЕТОДИ СТРУКТУРНО-ПАРАМЕТРИЧНОЇ ОПТИМІЗАЦІЇ ДИСКРЕТНИХ  
ВИРОБНИЧИХ ТЕХНОЛОГІЧНИХ ПРОЦЕСІВ

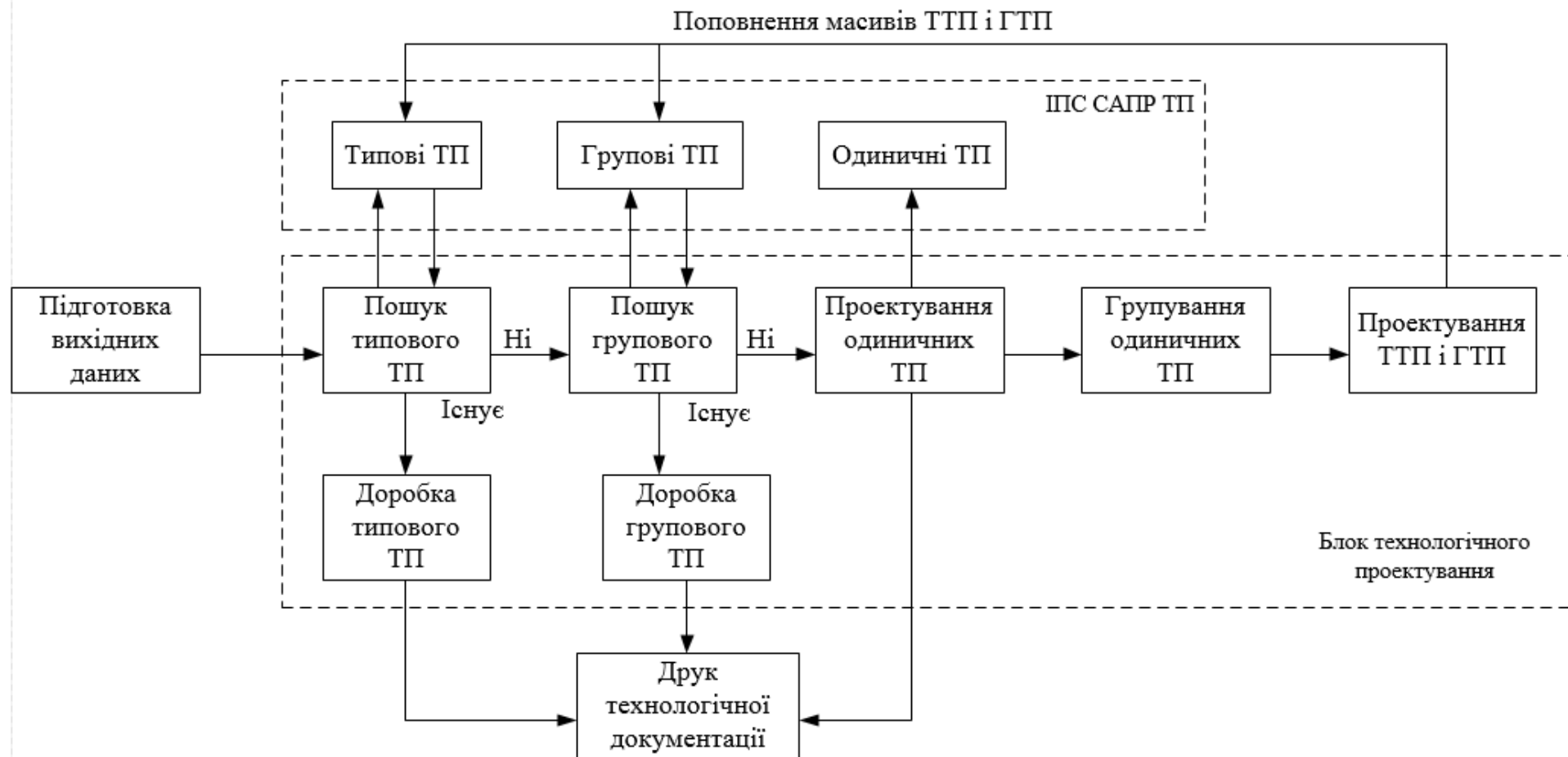
Графічний матеріал

ГЮИК.504300.014

АРКУШІВ 9

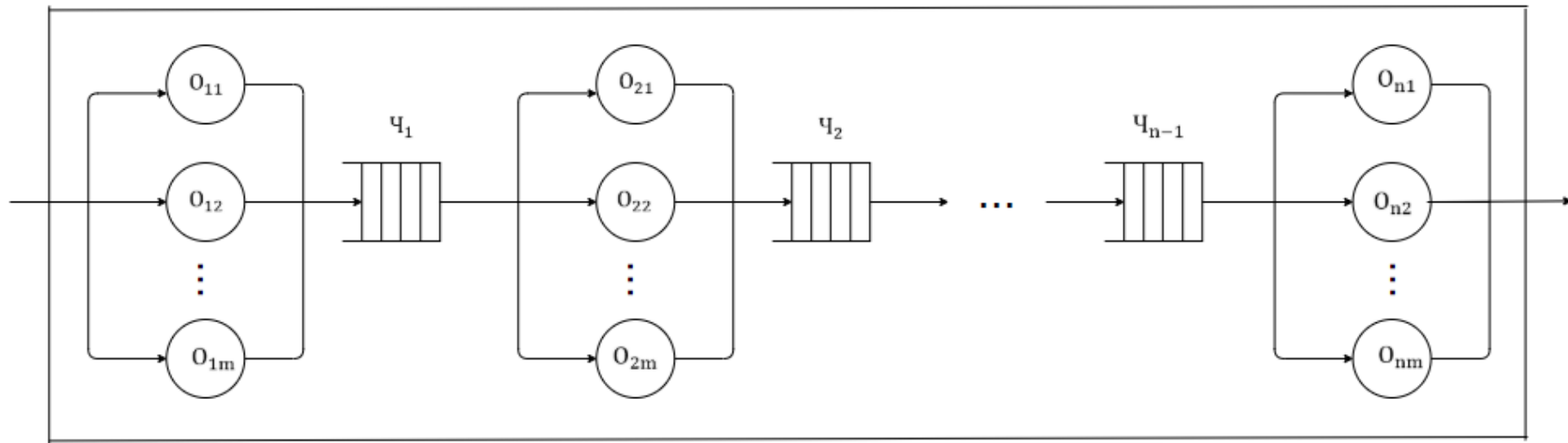
2020 р.

## УЗАГАЛЬНЕНА СТРУКТУРНА СХЕМА САПР ТП



Розробив	Сідорчук Є. І.			Методи структурно-параметричної оптимізації дискретних виробничих технологічних процесів	
Перевірив	Безкоровайний В.В.				
Н. контроль	Безкоровайний В.В.			СПРМ-18-2	Аркуш 1
Затвердив	Гребеннік І.В.			СТ	Аркушів 1

## СТРУКТУРНА СХЕМА ЛІНІЙНОГО ТЕХНОЛОГІЧНОГО ПРОЦЕСУ



$O_{ij}$  – обладнання, необхідне для виготовлення об'єкта (деталі);

$i$  – номер фази у технологічному процесі об'єкта,  $i = \overline{1, n}$ ;

$n$  – кількість фаз у технологічному процесі;

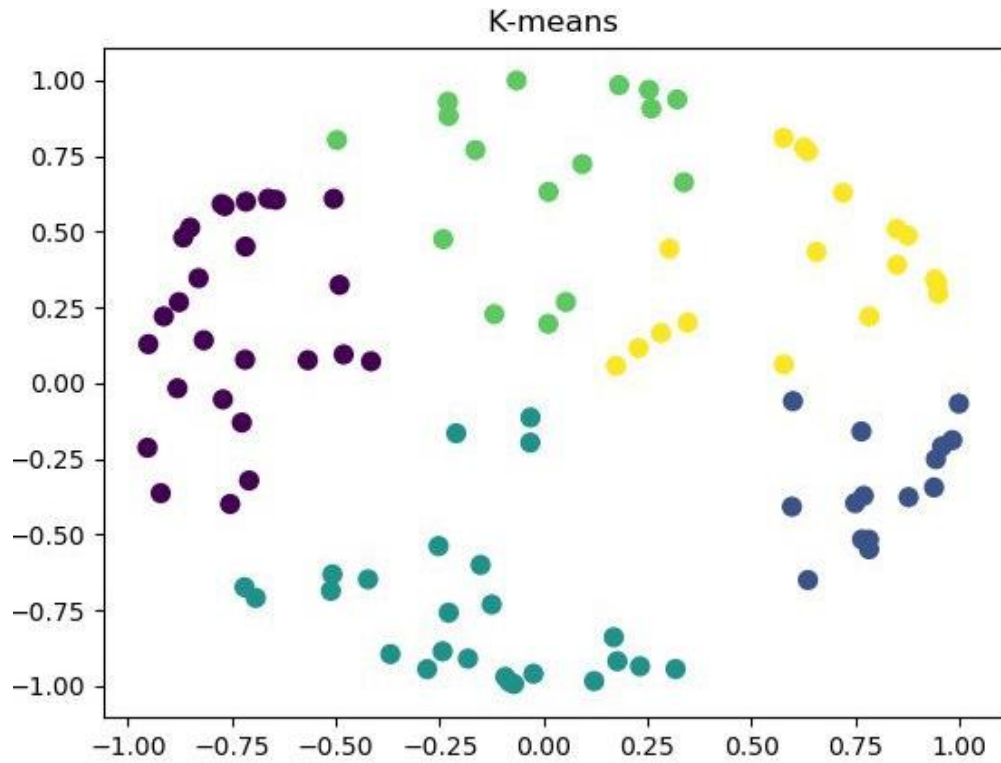
$j$  – номер обладнання в рамках фази технологічного процесу об'єкта,  $j = \overline{1, m}$ ;

$m$  – загальна кількість обладнання в рамках фази технологічного процесу об'єкта;

Ч – черги, що можуть утворюватися при переходами між фазами.

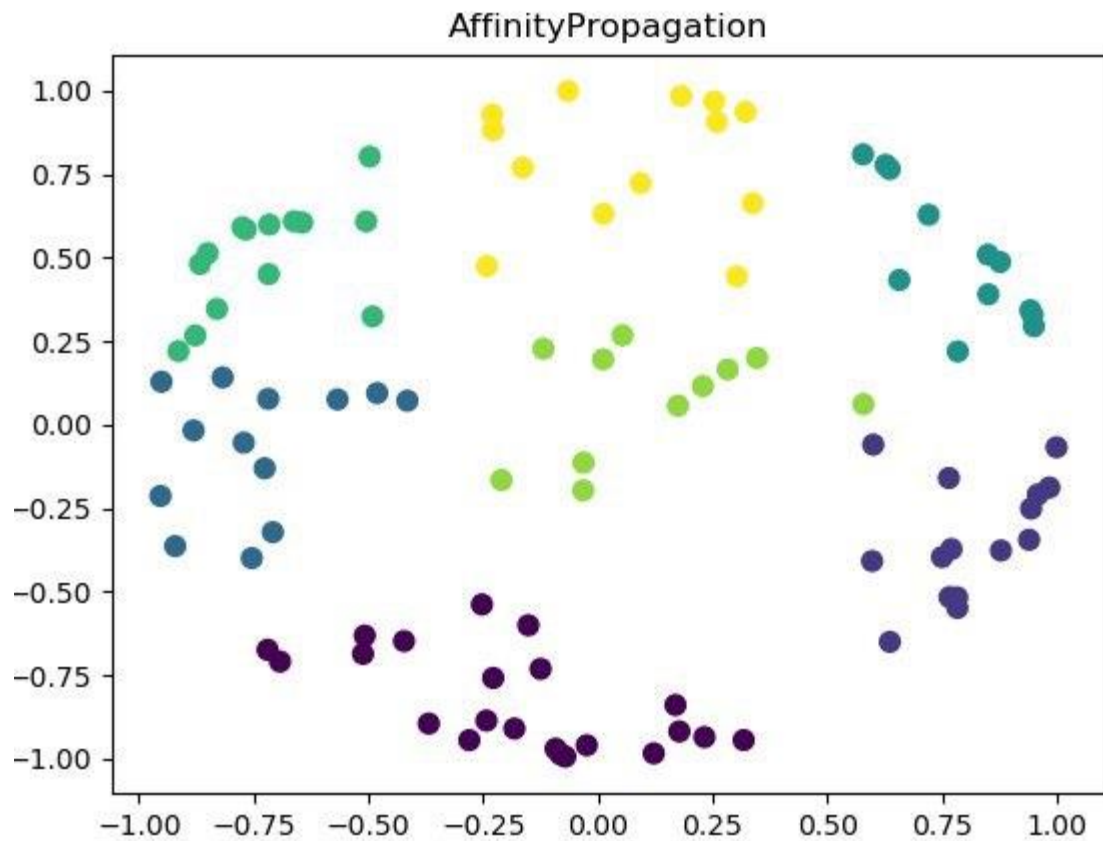
Розробив	Сідорчук Є. І.			<i>Методи структурно-параметричної оптимізації дискретних виробничих технологічних процесів</i>	
Перевірив	Безкоровайний В.В.				
Н. контроль	Безкоровайний В.В.			СПРМ-18-2	Аркуш 1
Затвердив	Гребеннік І.В.			СТ	Аркушів 1

## ПРИКЛАД РОЗПОДІЛУ ТП НА КЛАСТЕРИ (АЛГОРИТМ К-MEANS)



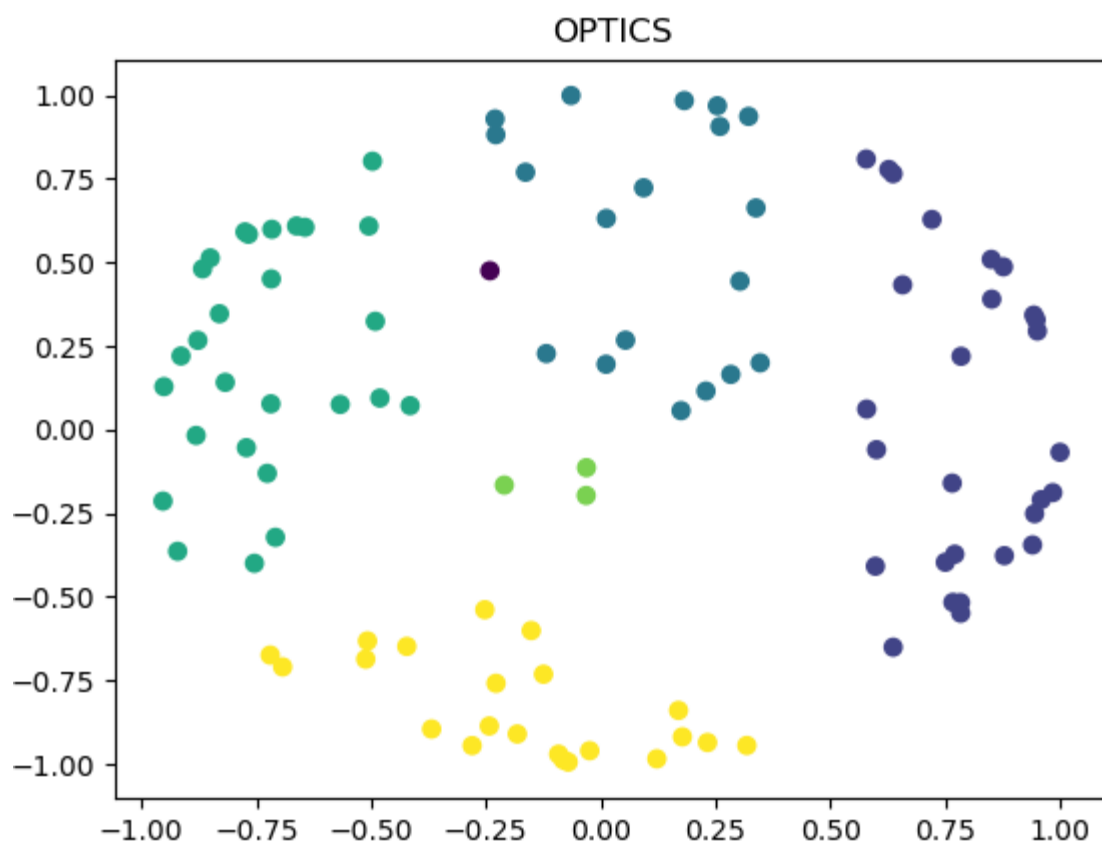
<i>Розробив</i>	<i>Сідорчук Є. І.</i>			<i>Методи структурно-параметричної оптимізації дискретних виробничих технологічних процесів</i>	
<i>Перевірів</i>	<i>Безкорвайний В.В.</i>				
<i>Н. контроль</i>	<i>Безкорвайний В.В.</i>			<i>СПРМ-18-2</i>	<i>Аркуш 1</i>
<i>Затвердив</i>	<i>Гребеннік І.В.</i>			<i>СТ</i>	<i>Аркушів 1</i>

**ПРИКЛАД РОЗПОДІЛУ ТП НА КЛАСТЕРИ  
(АЛГОРИТМ AFFINITY PROPAGATION)**



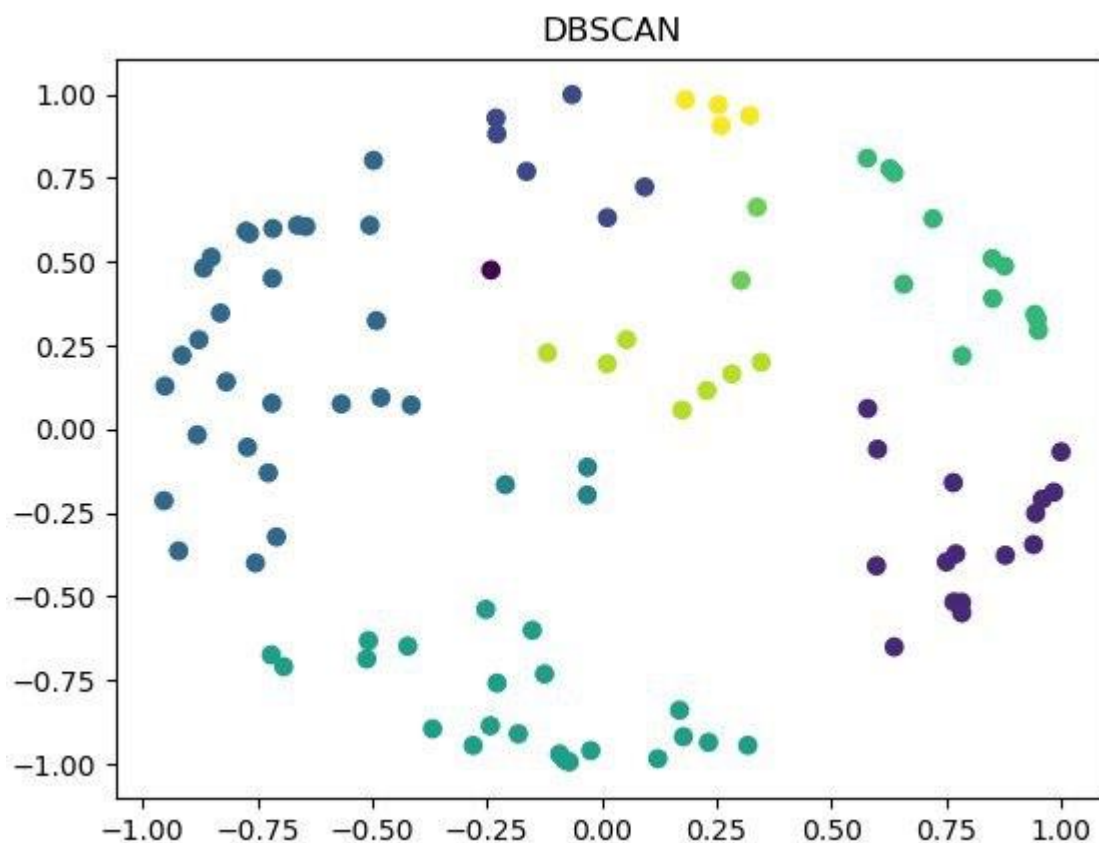
<i>Розробив</i>	<i>Сідорчук Є. І.</i>			<i>Методи структурно-параметричної оптимізації дискретних виробничих технологічних процесів</i>	
<i>Перевірив</i>	<i>Безкоровайний В.В.</i>				
<i>Н. контроль</i>	<i>Безкоровайний В.В.</i>			<i>СПРМ-18-2</i>	<i>Аркуш 1</i>
<i>Затвердив</i>	<i>Гребеннік І.В.</i>			<i>СТ</i>	<i>Аркушів 1</i>

## ПРИКЛАД РОЗПОДІЛУ ТП НА КЛАСТЕРИ (АЛГОРИТМ OPTICS)



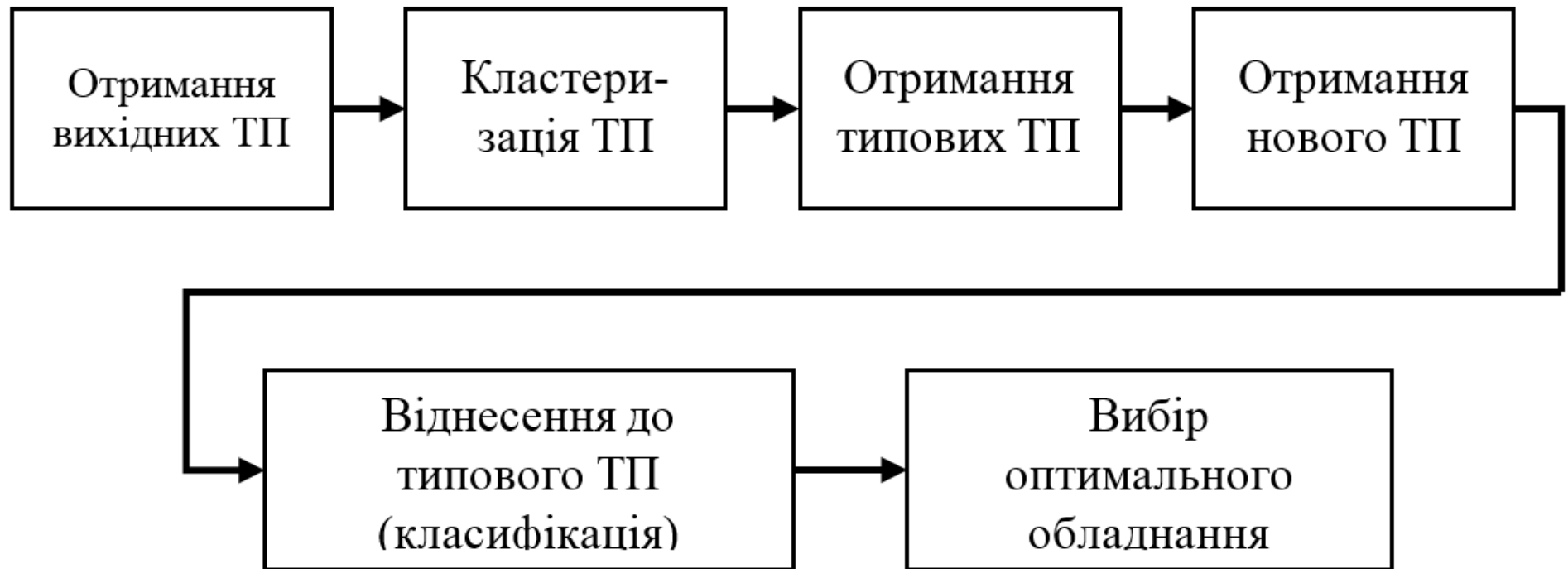
<i>Розробив</i>	<i>Сідорчук Є. І.</i>			<i>Методи структурно-параметричної оптимізації дискретних виробничих технологічних процесів</i>	
<i>Перевірів</i>	<i>Безкоровайний В.В.</i>				
<i>Н. контроль</i>	<i>Безкоровайний В.В.</i>			<i>СПРМ-18-2</i>	<i>Аркуш 1</i>
<i>Затвердив</i>	<i>Гребеннік І.В.</i>			<i>СТ</i>	<i>Аркушів 1</i>

## ПРИКЛАД РОЗПОДІЛУ ТП НА КЛАСТЕРИ (АЛГОРИТМ DBSCAN)



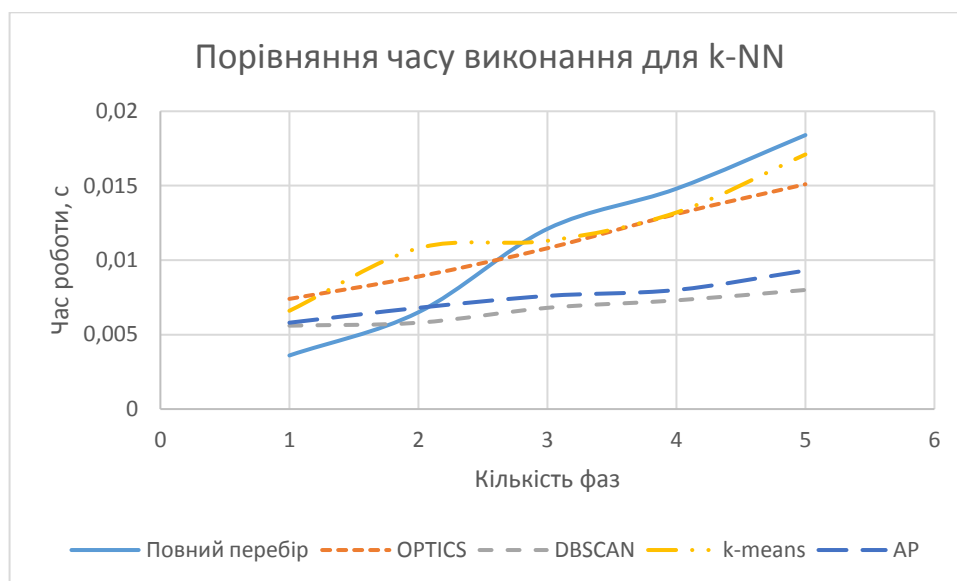
<i>Розробив</i>	<i>Сідорчук Є. І.</i>			<i>Методи структурно-параметричної оптимізації дискретних виробничих технологічних процесів</i>	
<i>Перевірів</i>	<i>Безкоровайний В.В.</i>				
<i>Н. контроль</i>	<i>Безкоровайний В.В.</i>			<i>СПРМ-18-2</i>	<i>Аркуш 1</i>
<i>Затвердив</i>	<i>Гребеннік І.В.</i>			<i>СТ</i>	<i>Аркушів 1</i>

## ЗАГАЛЬНА СХЕМА МЕТОДУ ОПТИМІЗАЦІЇ ВИРОБНИЧОГО ТП



<i>Розробив</i>	<i>Сідорчук Є. І.</i>			<i>Методи структурно-параметричної оптимізації дискретних виробничих технологічних процесів</i>	
<i>Перевірив</i>	<i>Безкорвайний В.В.</i>				
<i>Н. контроль</i>	<i>Безкорвайний В.В.</i>			<i>СПРМ-18-2</i>	<i>Аркуш 1</i>
<i>Затвердив</i>	<i>Гребеннік І.В.</i>			<i>СТ</i>	<i>Аркушів 1</i>

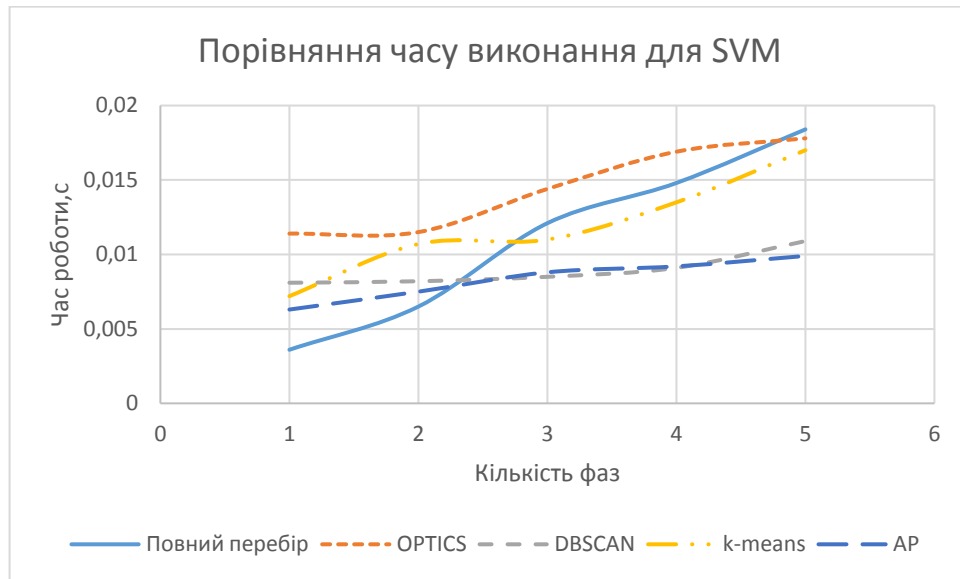
## ПОРІВНЯННЯ ЧАСУ ВИКОНАННЯ ДЛЯ АЛГОРИТМУ K-NN



Кількість фаз	1	2	3	4	5
Повний перебір	0,0036	0,0065	0,0121	0,0148	0,0184
OPTICS	0,0074	0,0089	0,0108	0,0131	0,0151
DBSCAN	0,0056	0,0058	0,0068	0,0073	0,008
k-means	0,0066	0,0108	0,0113	0,0132	0,0171
AP	0,0058	0,0068	0,0076	0,008	0,0093

Розробив	Сідорчук Є. І.			Методи структурно-параметричної оптимізації дискретних виробничих технологічних процесів	
Перевірив	Безкорвайний В.В.				
Н. контроль	Безкорвайний В.В.			СПРМ-18-2	Аркуш 1
Затвердив	Гребеннік І.В.			СТ	Аркушів 1

## ПОРІВНЯННЯ ЧАСУ ВИКОНАННЯ ДЛЯ АЛГОРИТМУ SVM



Кількість фаз	1	2	3	4	5
Повний перебір	0,0036	0,0065	0,0121	0,0148	0,0184
OPTICS	0,0114	0,0115	0,0144	0,0169	0,0178
DBSCAN	0,0081	0,0082	0,0085	0,0091	0,0109
k-means	0,0072	0,0107	0,011	0,0135	0,0117
AP	0,0063	0,0075	0,0088	0,0092	0,0099

Розробив	Сідорчук Є. І.			Методи структурно-параметричної оптимізації дискретних виробничих технологічних процесів	
Перевірив	Безкорвайний В.В.				
Н. контроль	Безкорвайний В.В.			СПРМ-18-2	Аркуш 1
Затвердив	Гребеннік І.В.			СТ	Аркушів 1

## ДОДАТОК Б

Текст програми

ГЮИК.504300.014-01 12 01

(позначення документу)

Харківський національний університет радіоелектроніки

«ЗАТВЕРДЖУЮ»  
керівник атестаційної роботи  
проф. Безкоровайний В.В.

МЕТОДИ СТРУКТУРНО-ПАРАМЕТРИЧНОЇ ОПТИМІЗАЦІЇ ДИСКРЕТНИХ  
ВИРОБНИЧИХ ТЕХНОЛОГІЧНИХ ПРОЦЕСІВ

OPTIMIZATION OF PRODUCTION PROCESSES

Текст програми

ЛИСТ ЗАТВЕРДЖЕННЯ

ГЮИК.504300.014-01 12 01

РОЗРОБИВ:  
ст. гр. СПРм-18-2  
Сідорчук Є. І.

2020 р

ГЮИК.505200.014 – 01 12 01

ЗАТВЕРДЖЕНО

ГЮИК.504300.014-01 12 01-ЛЗ

МЕТОДИ СТРУКТУРНО-ПАРАМЕТРИЧНОЇ ОПТИМІЗАЦІЇ ДИСКРЕТНИХ  
ВИРОБНИЧИХ ТЕХНОЛОГІЧНИХ ПРОЦЕСІВ

OPTIMIZATION OF PRODUCTION PROCESSES

Текст програми

ГЮИК.504300.004 – 01 12 01

АРКУШІВ 5

2020 р.



```

def exhaustive_search(test_item, items, n_stage):
    min = [[-1] for i in range(2)]
    index = [[-1] for i in range(2)]
    for i in range(n_stage):
        type = test_item.at[0, 'Type'][i]
        power = test_item.at[0, 'Power'][i]
        for j in range(items.index.size):
            if items.at[j, 'Type'] == type and items.at[j, 'Power'] >= power:
                if min[i] != -1 and items.at[j, 'Value'] > min[i]:
                    min[i] = items.at[j, 'Value']
                    index[i] = j
                elif min[i] == -1:
                    min[i] = items.at[j, 'Value']
                    index[i] = j
    return min, index

def get_centroids(labels, items, n_centroids):
    centroids = pd.DataFrame(columns=["Mi1", "Mi2", "Mi3", "Mi4", "Mi5", "Mi6", "Mi7"])
    for i in range(n_centroids + 1):
        mi = [[0] for i in range(7)]
        count = 0
        for j in range(len(labels)):
            if labels[j] == i:
                for k in range(7):
                    mi[k] += items.iat[j, k]
                count += 1
        centroids.loc[i] = {"Mi1": mi[0] / count, "Mi2": mi[1] / count, "Mi3": mi[2] / count, "Mi4": mi[3] /
count,
                            "Mi5": mi[4] / count, "Mi6": mi[5] / count, "Mi7": mi[6] / count}
    return centroids

def k_neighbors_classifier(list_centroids, test):
    classifier = KNeighborsClassifier(n_neighbors=3)
    x = list_centroids.iloc[:, :7].values
    y = list_centroids.index.values
    classifier.fit(x, y)
    KNN_prediction = classifier.predict(test)
    return KNN_prediction

def svm_classifier(list_centroids, test):
    x = list_centroids.iloc[:, :7].values
    y = list_centroids.index.values
    SVC_model = svm.SVC()
    SVC_model.fit(x, y)
    SVC_prediction = SVC_model.predict(test)
    return SVC_prediction

```

```
colors = ['royalblue', 'maroon', 'forestgreen', 'mediumorchid', 'tan', 'deeppink', 'olive', 'goldenrod',
'lightcyan',
    'navy', 'red', 'green', 'yellow',
    'orange', 'purple', 'blue', 'black']

vectorizer = np.vectorize(lambda x: colors[x % len(colors)])

test_item = np.random.sample((1, 7))
test_x = pd.DataFrame({'Mi1': np.random.uniform(0, 1, 1),
    'Mi2': np.random.uniform(0, 1, 1),
    'Mi3': np.random.uniform(0, 1, 1),
    'Mi4': np.random.uniform(0, 1, 1),
    'Mi5': np.random.uniform(0, 1, 1),
    'Mi6': np.random.uniform(0, 1, 1),
    'Mi7': np.random.uniform(0, 1, 1),
    "Type": [list(np.random.randint(1, 4, 2))],
    "Power": [list(np.random.uniform(1, 9, 2))]])

# generation_data(1000)
X = pd.read_excel('Data/items.xlsx', index_col=0)
data = X.iloc[:, :7].values

pca = PCA(n_components=2)
X_principal = pca.fit_transform(data)
X_principal = pd.DataFrame(X_principal)
X_principal.columns = ['P1', 'P2']
start_time = time.time()
db_default = DBSCAN(eps=0.345, min_samples=2).fit(data)
end_time = time.time()
print("DB: %s" % (end_time - start_time))
labels = db_default.labels_
# print(labels)
DB = X
DB['centroids'] = labels
DB.to_excel('DB.xlsx')
centroids = get_centroids(labels, X, max(labels))
# print(centroids)
start_time = time.time()
KNN_prediction = k_neighbors_classifier(centroids, test_x.iloc[:, :7].values)

min, index = min_value(labels, test_x, X, KNN_prediction, len(test_x.at[0, 'Type']))
end_time = time.time()
print("kNN: %s" % (end_time - start_time))
print(KNN_prediction)
print(min)
print(index)

start_time = time.time()
SVC_prediction = svm_classifier(centroids, test_x.iloc[:, :7].values)
```

```
min, index = min_value(labels, test_x, X, SVC_prediction, len(test_x.at[0, 'Type']))
end_time = time.time()
print("SVM: %s" % (end_time - start_time))
print(SVC_prediction)
print(min)
print(index)
start_time = time.time()
kmeans = KMeans(n_clusters=int(X.index.size * 0.2)).fit(data)
end_time = time.time()
print("Kmeans: %s" % (end_time - start_time))
k_means = copy.deepcopy(X)
k_means['centroids'] = kmeans.labels_
k_means.to_excel('Data/k_means.xlsx')
centroids = pd.DataFrame(kmeans.cluster_centers_)
start_time = time.time()
KNN_prediction = k_neighbors_classifier(centroids, test_x.iloc[:, :7].values)

min, index = min_value(kmeans.labels_, test_x, X, KNN_prediction, len(test_x.at[0, 'Type']))
end_time = time.time()
print("kNN: %s" % (end_time - start_time))
print(KNN_prediction)
print(min)
print(index)

start_time = time.time()
SVC_prediction = svm_classifier(centroids, test_x.iloc[:, :7].values)

print(SVC_prediction)
min, index = min_value(kmeans.labels_, test_x, X, SVC_prediction, len(test_x.at[0, 'Type']))
end_time = time.time()
print("SVM: %s" % (end_time - start_time))
print(min)
print(index)

clust = OPTICS(min_samples=2)
print("OPTICS")
# Run the fit
start_time = time.time()
clust.fit(data)
end_time = time.time()
print("OPTICS: %s" % (end_time - start_time))
labels = clust.labels_[clust.ordering_]
centroids = get_centroids(labels, X, max(labels))
start_time = time.time()
KNN_prediction = k_neighbors_classifier(centroids, test_x.iloc[:, :7].values)

print(KNN_prediction)
min, index = min_value(labels, test_x, X, KNN_prediction, len(test_x.at[0, 'Type']))
end_time = time.time()
print("kNN: %s" % (end_time - start_time))
```

```
print(min)
print(index)

start_time = time.time()
SVC_prediction = svm_classifier(centroids, test_x.iloc[:, :7].values)

print(SVC_prediction)
min, index = min_value(labels, test_x, X, SVC_prediction, len(test_x.at[0, 'Type']))
end_time = time.time()
print("SVM: %s" % (end_time - start_time))
print(min)
print(index)
start_time = time.time()
af = AffinityPropagation().fit(data)
end_time = time.time()
print("AffinityPropagation: %s" % (end_time - start_time))
cluster_centers_indices_ = af.cluster_centers_indices_
labels = af.labels_
affinityPropagation = X
affinityPropagation['centroids'] = labels
affinityPropagation.to_excel('AffinityPropagation.xlsx')
centroids = pd.DataFrame(af.cluster_centers_)
start_time = time.time()
KNN_prediction = k_neighbors_classifier(centroids, test_x.iloc[:, :7].values)

print(KNN_prediction)

min, index = min_value(kmeans.labels_, test_x, X, KNN_prediction, len(test_x.at[0, 'Type']))
end_time = time.time()
print("kNN: %s" % (end_time - start_time))
print(min)
print(index)
start_time = time.time()
SVC_prediction = svm_classifier(centroids, test_x.iloc[:, :7].values)

print(SVC_prediction)
min, index = min_value(kmeans.labels_, test_x, X, SVC_prediction, len(test_x.at[0, 'Type']))
end_time = time.time()
print("SVM: %s" % (end_time - start_time))
print(min)
print(index)

start_time = time.time()
min, index = exhaustive_search(test_x, X, len(test_x.at[0, 'Type']))
end_time = time.time()
print("ES: %s" % (end_time - start_time))
print(min)
print(index)
```

## ДОДАТОК В

## Сертифікат учасника конференції



# СЕРТИФІКАТ

№ К-140 виданий 23 квітня 2020 року

засвідчує, що

**Єлизавета Іванівна СІДОРЧУК**

взяв(ла) активну участь у роботі  
V Міжнародної науково-практичної конференції

**Інформаційні технології в культурі, мистецтві, освіті, науці, економіці та бізнесі**

21-22 квітня 2020 року,  
м. Київ, Україна

Від імені Оргкомітету  
**Олена ЧАЙКОВСЬКА**

ІНСТИТУТ МОДЕРНІЗАЦІЇ ЗМІСТУ ОСВІТИ

VYTAUTAS MAGNUS UNIVERSITY, LITHUANIA

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КУЛЬТУРИ І МИСТЕЦТВ

DANUBIUS UNIVERSITY, GALATI, ROMANIA

УКРАЇНЬКА ФЕДЕРАЦІЯ ІНФОРМАТИКИ

## ДОДАТОК Г

Відомість атестаційної роботи

---

ГЮИК.504300.014

(Позначення документу)

