

汽車裝配線多載量小車即時調度

學生：胡理嫻 108034466 指導教授：邱銘傳教授

國立清華大學 工業工程與工程管理學系

摘要：在製造系統中，多載量小車被廣泛應用於物料搬運過程中。優化即時多載量小車調度不僅能夠實現更高的生產率也能獲得更好的製造系統性能。因此，為了有效地解決汽車裝配線中多載量小車的即時調度問題，本文提出了基於支持向量機（support vector machine, SVM）、基於卷積神經網路（convolutional neural network, CNN）的即時調度方法。首先對多載量小車的即時調度問題進行描述。然後在 eM-Plant 軟體中建立了模擬模型，通過裝配線物料搬運系統模擬生成的樣本，離線訓練 CNN 與 SVM 模型，並在即時階段利用 CNN 與 SVM 實現多載量小車“等待”或“搬運”的調度決策。實驗結果表明，基於 SVM 的調度機制能夠動態地對多載量小車進行即時優化決策，在適應性、預測準確性和泛化性能上均優於 CNN。

關鍵字：即時調度、SVM、CNN、物料搬運系統、多載量小車

1. 引言

目前，汽車製造業大多採用具有較高搬運能力的多載量小車(dolly-train)進行物料搬運[1]。因為汽車裝配系統本身極具複雜性，通過提升物料搬運系統中的多載量小車即時調度水準來提高汽車裝配線生產率已經成為實現汽車製造業提高效率、降低成本的重要手段。

物料搬運系統調度問題具有動態性、隨機性等特點。但是，已有的關於動態調度的研究主要集中於基於規則的調度方法[1]。例如，Egbelu 等人[2]採用了最短搬運時間先服務規則進行即時調度。與之類似，Sabuncuoglu 等人[3]採用最長輸出佇列規則實現動態調度。Wu 等[4]提出了基於自我調整多參數的調度策略來解決動態調度問題。目前，也有一些學者通過模擬、改進的啟發式演算法、機器學習等方法優化物料搬運系統動態調度。如 Hsieh 等[5]將物料搬運系統細分為多個不重疊的、包含自動引導小車的子系統，然後運用模擬方法獲取小車的最佳數量及最優資源配置。Godinho 等[6]提出了一個分類系統，圍繞作業描述、資源約束類型、解決方法等 6 個方面，對運用遺傳演算法解決柔性製造系統調度問題進行研究。隨著對單載量搬運設施的研究日趨完善，與多載量小車相關的研究也愈加受到學術界的重視。如 Chen 等[7]研究了在汽車裝配線上受後進先出約束的多載量小車調度問題，基於 ANN 的匹配器和知識庫，比較了不同搬運任務選擇和排序規則。Fathi 等[8]同時最小化搬運次數和線邊庫存，提出了改進的模擬退火演算法來優化多載量小車每次配送的零件種類和數量。但是，已有文獻均專注於研究多載量小車的資源配置與任務排序問題，鮮有研究涉及其動態調度決策問題。

因此，本文引入支援向量機與卷積神經網路，並結合模擬技術，針對汽車裝配線中多載量小車的動態調度決策問題，提出了基於 SVM 的多載量小車即時調度方法與基於 CNN 的多載量小車即時調度方法。

2. 問題描述

汽車裝配線物料搬運系統將裝配線分為若干個裝配區域，每個裝配區域由若干輛多載量小車負責其所有工作站的物料配送。一個物料搬運區域活動邏輯圖如圖 1 所示。

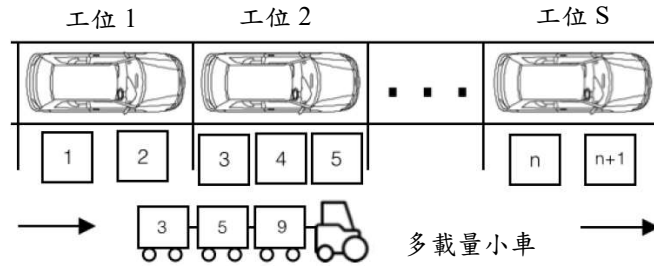


圖1.裝配線的一個物料搬運區

為有效說明該調度問題，本文將整個物料搬運系統狀態記為 X_k ，系統狀態是在決策時間點 t_k 時各個工位元的線邊庫存狀態 BF_i 與當前庫存可維持裝配的時長 t_i^l ($i=1,2,\dots,p$) 的集合，所以有：

$$X_k = \{BF_1, BF_2, \dots, BF_p, t_1^l, t_2^l, \dots, t_p^l\}$$

把在各個決策點 t_k 的調度決策記為變數 Y_k ，用“0”表示“等待”，“1”表示“搬運”，則多載量小車的即時調度問題可轉化為一個“0-1”資料的二分類問題，即：

$$Y_k = W(X_k) = \begin{cases} 0, & \text{等待} \\ 1, & \text{搬運} \end{cases}$$

3. 研究方法

本文首先使用eM-plant模擬汽車裝配線多載量小車搬運決策過程，獲取搬運系統狀態資料，包括各個工位元在決策時間點時線邊庫存狀態、當前庫存可維持裝配的時長、多載量小車的搬運決策。根據這些資料，對CNN與SVM模型分別進行訓練，然後使用已經訓練好的模型對汽車裝配線的多載量小車進行即時物料搬運決策。

3.1 eM-plant 模擬

根據上節問題描述與參照文獻[1]中物料搬運系統模擬建模方法，建立模擬模型如圖2所示。模擬過程的主要資料存儲於BF_T、Lack_T與Decision表中，模擬結束後，編寫代碼將所有資料整合到Data表中，具體如圖3所示。

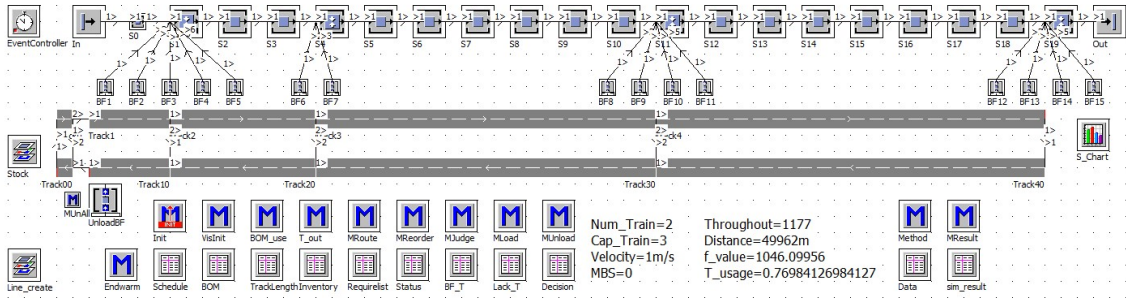


圖2. 模擬系統模型

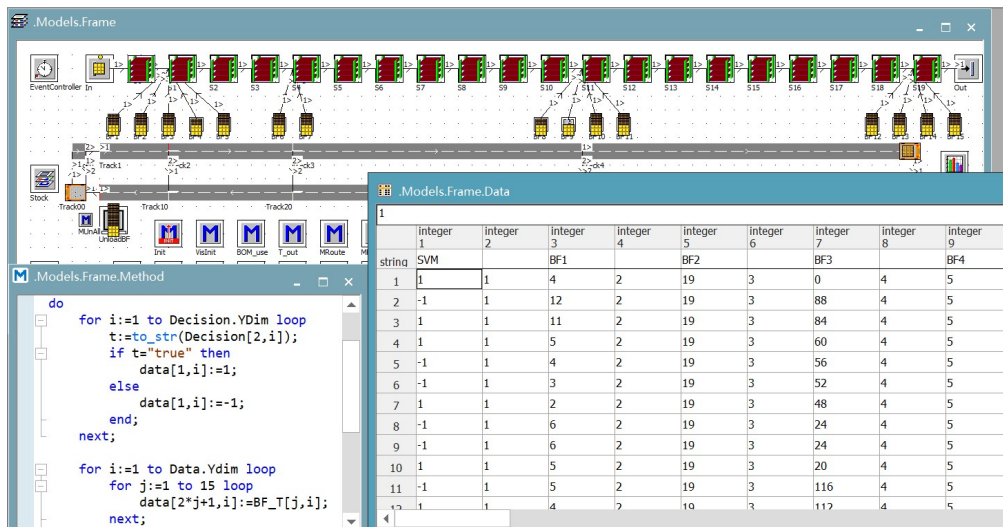


圖3 資料收集整合代碼與資料表

3.2 數據預處理

3.2.1 資料過抽樣與歸一化

經由分析發現，獲取的原始資料是不平衡的（imbalanced），因此採用SMOTE方法對訓練集進行過抽樣以保證資料平衡，代碼及結果如圖4所示。圖中，Counter裡的資料為原始資料的統計分析，“0”類別的樣本數為44476筆，“1”類別的樣本數為13855筆，“1”類樣本數過少。dict_items給出了過抽樣後兩類樣本的數量，可知，最終用於訓練的兩類樣本數都是44476筆。

```

from imblearn.over_sampling import SMOTE
from collections import Counter
X=train_images
y=train_labels

print(Counter(y))

X_resampled_a, y_resampled_a = SMOTE().fit_sample(X, y)

print(Counter(y_resampled_a).items())

```

```

Counter({0: 44476, 1: 13855})
dict_items([(1, 44476), (0, 44476)])

```

圖4 過抽樣代碼與結果

為消除奇異樣本資料導致的不良影響，將所有原始資料數值範圍限定在[0,1]之間。本文分別採用線性歸一化、標準化與非線性歸一化三種方式進行資料預處理，結果如圖5所示。因此，本文選取線性歸一化用於資料預處理。

	precision	recall	f1-score	support		precision	recall	f1-score	support		precision	recall	f1-score	support
-1	0.9537634409	1.0000000000	0.9763346175	8870	-1	0.7700733075	0.9988813066	0.8696795559	8939	-1	0.7633901791	1.0000000000	0.8658210624	8908
1	1.0000000000	0.8463737049	0.9167956656	2799	1	0.8648648649	0.0234432234	0.0456490728	2730	1	0.0000000000	0.0000000000	0.0000000000	2761
accuracy	0.9631502271 11669				accuracy	0.7706744365 11669				accuracy	0.7633901791 11669			

線性歸一化

標準化

非線性歸一化

圖5 三種歸一化結果對比

3.2.2 特徵選擇

本文選擇相關性分析與隨機森林 (random forest, RF) 選取重要特徵，以避免過度擬合和維度災難。兩種方法獲取的特徵排名匯總於表1，數字1~15表示 $BF_1 \sim BF_{15}$ ，數字16~30表示 $t_1^1 \sim t_{15}^1$ 。由表知，重要特徵為 BF_1 與 t_{14}^1 。相關性分析方法是為了過濾出並只獲取相關特徵的子集。其中，過濾使用相關矩陣完成，通常使用Pearson Correlation進行。在相關矩陣中，相關係數的值在-1和1之間：接近0的值表示弱相關（精確0表示沒有相關），接近1的值表示強正相關，而接近-1的值表示強烈的負相關。首先繪製一個Pearson相關熱圖（如圖6所示），然後看一下獨立變數與輸出變數MEDV之間的相關性。分別選擇與輸出變數的相關性大於0.005與0.01的特徵，選出前21與前23個特徵分別用於訓練。在資料科學工作中，RF是一種常用的特徵選擇方法。特徵重要性評估的思想是查看每個特徵對隨機森林中的每棵樹有多少貢獻，然後取平均值，最後比較特徵之間的貢獻比率。通常可以使用基尼係數或OOB錯誤率作為評估指標來衡量貢獻。本文選擇從scikit-learn隨機森林類庫中調用“RandomForestClassifier()”以對特徵進行過濾和排序。本文研究分別選取RF的前16與前26個特徵用於模型訓練。但是RF在特徵選擇過程中因為bagging抽取子集的時候，可能具有一定的隨機性，決策樹根據gini分出的結果有差異，即使用RF兩次進行特徵篩選獲取的特徵排名略有差異，如表1中RF-26與RF-16所示。

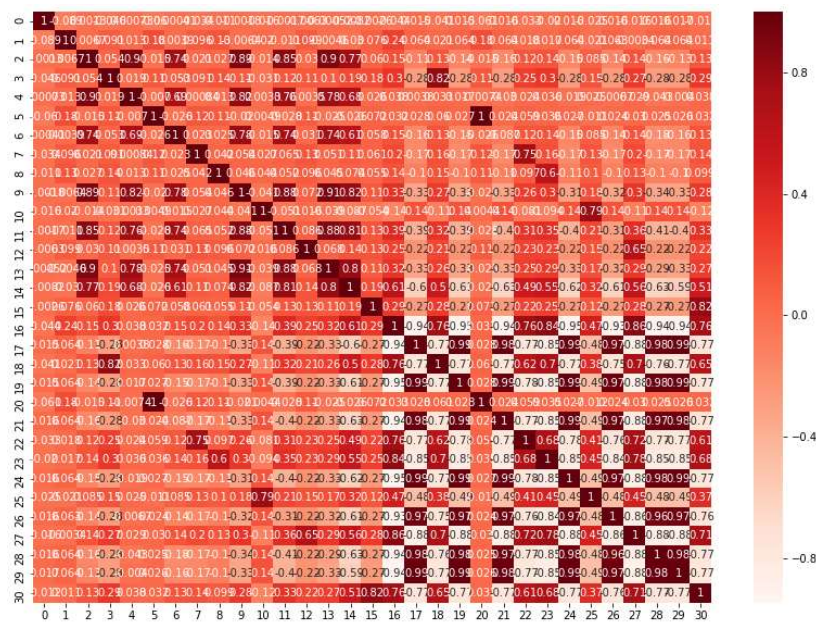


圖6 Pearson相關熱圖

表1 特徵排名

排名序號	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
RF-26	16	1	20	27	23	22	3	18	25	30	12	8	5	15	7	10	2	6	13	11	4	14	9	19	28	21	17	24	29	26
RF-16	16	1	20	27	3	23	18	25	30	22	5	7	8	15	12	10	2	4	9	11	13	6	14	19	17	24	29	21	28	26
Correlation	1	20	5	3	16	18	7	22	25	23	10	26	24	21	19	17	8	14	12	15	9	11	2	4	13	6	27	28	29	30

3.3 卷積神經網路

CNN是一種前饋神經網路，是多層感知機的變種。它的人工神經元可以回應一部分覆蓋範圍內的周圍單元，對於大型影像處理有出色表現。它包括卷積層和池化層。在卷積層，通過卷積運算提取出圖像的特徵，可以使得原始信號的某些特徵增強，並且降低雜訊。一般用一個可訓練的filter去卷積一個輸入的圖像（第一階段是輸入的圖像，後面的階段就是卷積特徵map了），然後加一個偏置得到卷積層。然後，對圖像進行pooling操作，可以減少資料處理量同時保留有用資訊，採樣可以混淆特徵的具體位置，因為某個特徵找出來之後，它的位置已經不重要了，只需要這個特徵和其他特徵的相對位置，可以應對形變和扭曲帶來的同類物體的變化。例如，每鄰域四個圖元求和變為一個圖元，然後加權、增加偏置，最後通過一個sigmoid激活函數，產生一個大概縮小四倍的特徵映射圖。圖7為整個CNN結構示意圖。

CNN具有局部連結與參數共用這兩個特點。局部連接是指卷積層輸出矩陣上的某個位置只與部分輸入矩陣有關，而不是全部的輸入矩陣。卷積層輸出的某個特徵可能只和輸入圖片的某一部分相關，和其它位置的資訊沒有任何關聯，局部連接可以讓特徵只關注其應該關注的部分。同時也減少了神經網路的參數。參數共用是指同一卷積層中 filter 的參數是共用的，一個 filter 無論在哪個位置進行卷積操作，filter 矩陣中的值都是一樣的（當然同一層不同的 filter 參數不一樣，不同層之間 filter 的參數也不一樣）。共用 filter 的參數可以使得圖像中的內容不受位置的影響。以MNIST手寫數位識別為例，無論數位“1”出現在左上角還是右下角，圖片的種類的都是不變的。共用卷積層 filter 的參數還可以極大地減少神經網路上的參數。

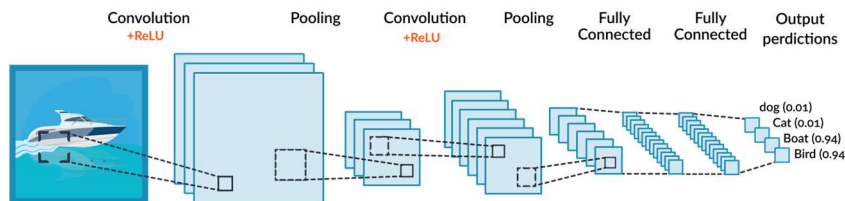


圖7 CNN結構示意圖

3.4 支持向量機

SVM是一種機器學習方法，其具有理論嚴密、適應性強、全域優化、訓練效率高和泛化性能好等優點，並在一定程度上克服了維數災難和過學習等困難。在多載量小車的即時調度過程中，調度系統只需判斷在決策點時的最佳決策是“等待”或“搬運”，故採用SVM模型對多載量小車的即時狀態進行搬運決策。

SVM是用於分類、回歸和離群值檢測的一組監督學習方法。在分類中，SVC，NuSVC和LinearSVC是能夠對資料集執行多類分類的支持向量機方法。因此本文選取SVC求解多載量小車動態調度決策這種二分類問題。一般而言，在二維線性可分離資料的情況下，如圖8左側所示，一種典型的機器學習演算法是試圖找到一種邊界，以將資料劃分為最小錯誤分類錯誤。如果仔細觀察圖8左側，發現會有多個邊界正確地劃分了資料點。SVM與其他分類演算法的不同之處在於，它選擇決策邊界以最大程度地增加與所有類別的最近資料點之間的距離。SVM不僅可以找到決策邊界，還可以找到最佳的決策邊界。最佳決策邊界是從所有類別的最近點開始具有最大餘量的邊界。如圖8右側所示，距離決策邊界最近的、將決策邊界和這些點之間的距離最大化的點稱為支持向量。在支援向量機的情況下，決策邊界稱為最大餘量分類器或最大餘量超平面。

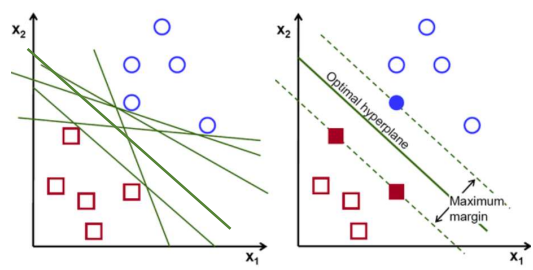


圖8 支持向量的決策邊界

本文調度決策問題屬於非線性可分的問題，即其特徵空間存在超曲面 (hypersurface) 將正類和負類分開。使用非線性函數可以將這類非線性可分問題從原始的特徵空間映射至更高維的希爾伯特空間 (Hilbert space)，從而轉化為線性可分問題。由於映射函數具有複雜的形式，難以計算其內積，可使用核方法 (kernel method)，即定義映射函數的內積為核函數 (kernel function)。因此，本文選取徑向基 (RBF) 內核去訓練SVM，其中C懲罰係數和gamma為兩個重要參數，該處均採用預設值設置。參數C，在所有SVM內核通用，與決策表面的簡單性相抗衡，可以對訓練樣本的誤分類進行有價轉換。較小的C會使決策表面更平滑，同時較高的C旨在正確地分類所有訓練樣本。gamma定義了單一訓練樣本能起到多大的影響。較大的gamma會更讓其他樣本受到影響。

4. 案例分析

4.1 模擬模型構建

裝配線生產三種不同的產品型號 (M1、M2、M3)，佈局如圖2所示。表1列出了每種產品型號所需的不同類型和數量的零件，以及每種零件料箱容量 SPQ_i 。在任何兩個連續的網站之間不考慮加工中的影響。裝配線的週期時間為85 s，每個工位的處理時間服從三角形分佈，下限為80 s，上限為90 s，模式為85 s。連接這些網站和中央倉庫的引導路徑是單向的。車站與中心倉庫的距離見表2最後一列。設置多載量小車的最佳數目、容量和速度值分別為2、3、1。此外，搬運人員完成裝載操作需要35秒，完成卸載操作需要45秒或130秒，記為LT=35秒，UT=45秒，

UT'=130秒。最後，在eM-plant程式設計中實現控制策略以協調裝配線和零件補充，將70000筆資料記錄在表中以用於CNN與SVM模型訓練及測試。

表2 BOM、 SPQ_i 與 Dis_i

P_i	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}
M_1	1	0	4	0	1	0	1	1	0	2	0	1	0	0	2
M_2	0	1	0	1	1	2	0	0	1	0	1	0	1	2	0
M_3	0	1	4	1	0	2	0	0	1	2	0	1	0	2	0
SPQ_i	10	22	96	16	22	44	18	14	16	70	34	12	24	33	33
Dis_i	12	12	12	12	12	30	30	72	72	72	72	120	120	120	120

4.2 CNN 模型構建與訓練驗證

構建CNN模型如圖9所示。通過一次變換一因數調整模型中的超參數水準設定，以提高CNN模型性能。表3給出了所有超參數組合的實驗結果。從表可知，當CNN模型的超參數設定值為表中黑體字的水準時，模型分類的準確性達到最高，即模型參數為：卷積3層、池化3層、激活函數為relu、優化器為rmsprop、訓練次數20次、訓練批大小為32、測試資料集大小占比16.67%。

```

model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(30, 32, 1)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(32, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))

model.add(layers.Flatten())
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(2, activation='softmax'))

```

圖9 CNN模型結構

圖10為CNN的訓練代碼。其中，反覆運算次數設置為20次，訓練集與測試集比例為5:1，從訓練集中隨機抽取12%的資料作為驗證集用於在訓練的時候調整模型參數。表4總結了使用全部資料、兩種特徵選擇後的資料對CNN訓練與測試的結果。從表4可知，使用Pearson correlation方法選擇23個特徵時，模型性能最好，比原始模型提高了6.81%，其泛化能力更好。因此，在工廠推薦使用相關性獲得的前23個特徵，來實現基於CNN的多載量小車動態調度決策。

```

from keras.callbacks import ModelCheckpoint

checkpoint = ModelCheckpoint(filepath='MLP_weights.best.hdf5', verbose=1, save_best_only=True)
rec = model.fit(train_images, train_labels, batch_size=32, epochs=20, validation_data=(x_valid, y_valid), callbacks=[checkpoint],
               verbose=2, shuffle=True)

model.load_weights('MLP_weights.best.hdf5')

```

圖10 CNN訓練代碼

表3 參數調整實驗結果與對應CNN模型性能

No.	Conv2D	MaxPooling2D	Activation Function	Learning Rate	Optimizer	Epoch	Train_batch	Test_data	Accuracy
1	1	1	relu	default	adamax	20	32	0.3	0.738905
2	1	1	sigmoid	default	adamax	20	32	0.3	0.706714
3	1	1	relu	default	adamax	20	32	0.3	0.765238
4	1	1	relu	default	adamax	5	32	0.3	0.738286
5	1	1	relu	default	sgd	5	32	0.3	0.693143
6	1	1	relu	default	rmsprop	5	32	0.3	0.777857
7	1	1	relu	default	adagrad	5	32	0.3	0.650619
8	1	1	relu	0.001	rmsprop	5	32	0.3	0.776238
9	1	1	relu	0.008	rmsprop	5	32	0.3	0.785571
10	1	1	relu	0.008	rmsprop	5	32	0.3	0.780667
11	1	1	relu	0.008	rmsprop	5	32	0.3	0.781095
12	1	1	relu	0.008	rmsprop	5	32	0.1	0.748089
13	1	1	relu	0.400	rmsprop	20	32	0.1667	0.806143
14	1	1	relu	0.600	rmsprop	20	32	0.1667	0.791571
15	1	1	relu	0.400	rmsprop	50	32	0.1667	0.805857
16	3	3	relu	default	rmsprop	20	32	0.1667	0.817465

表4 特徵選擇前後CNN分類準確性

	All features	RF-26	RF-16	Correlation-23	Correlation-21
Accuracy	0.817465	0.817808	0.812923	0.873168	0.838718

4.3 SVM 模型構建與訓練驗證

SVM模型構建如圖11所示。與CNN模型性能優化方法類似，通過實驗逐次調整SVM模型中的參數水準設定。表5給出了所有實驗結果。從表中可知，當SVM模型的參數設定值與資料處理方法為表中黑體部分時，模型分類的準確性達到最高，即選擇資料的16.67%作為測試集，不使用k-fold交叉驗證、歸一化與過抽樣。

```

from sklearn.svm import SVC
svclassifier = SVC(kernel='rbf')
svclassifier.fit(X_train, y_train)

y_pred = svclassifier.predict(X_test)

from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred, digits=10))

```

圖11 SVM模型結構

表5 超參數調整與對SVM應模型性能

k-fold	k	Test set	Normalization	Oversampling	Accuracy
no	-	0.1667	no	no	0.96417859
yes	10	0.1667	no	no	0.96165938
yes	5	0.1667	no	no	0.96165823
no	-	0.1000	no	no	0.96042857
no	-	0.1667	yes	no	0.96375011
yes	2	0.1667	yes	no	0.96165823
no	-	0.1667	no	yes	0.87628571
no	-	0.3000	no	no	0.87466667

設置訓練集與測試集比例為5:1。表6總結了使用全部資料、兩種特徵篩選後的資料對SVM訓練與測試的結果。從表中可知，SVM特徵選擇後分類準確幾乎不變，因此在工廠使用SVM進行多載量小車動態調度時，可以選取RF獲得的前16個特徵進行調度決策，以提高搬運系統效率。

表6 特徵選擇前後SVM分類準確性

	All features	RF-26	RF-16	Correlation-23	Correlation-21
Accuracy	0.9641786	0.9614363	0.9625503	0.9625503	0.9629788

為降低資料使用量，抽取20000筆資料進行訓練。表7總結了不同資料量下SVM的準確率。可知，20000資料訓練得到的SVM準確率達94.24%，仍然滿足裝配線調度需求；在k-fold交叉驗證後，SVM決策準確率高達97.40%，比70000筆資料訓練得到的模型的正確率還要略高。因此，SVM可使用較少的資料保證多載量小車決策調度的準確性，適用於裝配線資料收集困難或時間緊迫的情況。

表7 不同資料量下SVM的分類準確性

data amount	k-fold	k	Accuracy
70k	no	-	0.96417859
70k	yes	10	0.96165938
20k	no	-	0.94241152
20k	yes	80	0.97395722

4.4 實驗對比分析

結合表4與表6實驗資料可知，當樣本資料為70000筆時，SVM實現多載量小車動態調度的準確率最高達96.42%，平均單次決策反應時間0.000921s；CNN決策準確率最高為87.32%，平均單次決策反應時間0.000392s。當資料量減少到20000筆時，SVM分類準確率提高到97.40%。對比可知，SVM的準確率遠高於CNN，其反應時間也滿足生產調度需要。而且，在工廠無法收集

較多的資料量的情況下，SVM仍然能保證決策的正確性。因此，在穩定的生產環境下，基於SVM的多載量即時調度機制更適用於實際的生產調度。

5. 結論與展望

本文針對為汽車裝配線中多載量小車的動態調度問題，提出了基於支持向量機、基於卷積神經網路的多載量小車即時調度方法。首先使用 eM-Plant 軟體建立了模擬模型，通過混流裝配線的物料搬運系統模擬生成 70000 筆樣本，然後選取 58331 筆樣本對 CNN 與 SVM 離線訓練並用剩餘的 11669 筆樣本測試兩種模型在即時階段調度決策的準確性。實驗結果表明，基於 SVM 的即時調度方法不僅準確率高達 96.42%，正確性遠高於 CNN 的 87.32%；而且，基於 SVM 的即時調度方法的單次決策反應速度僅為 0.000921s 滿足即時調度需求。實驗也證明了，SVM 可使用較少的資料保證多載量小車決策調度的準確性，適用於裝配線資料收集困難或時間緊迫的情況。因此 SVM 較 CNN 更適用於動態物料搬運系統的多載量小車即時調度決策。另外，本文使用 Pearson 相關性與 RF 進行特徵篩選，將 30 個特徵縮減為 RF 前 16 個特徵，提高了 SVM 動態決策效率。

但是，本文所使用的資料量小於 100000，對於 CNN 訓練來說樣本數量過少，因此考慮在後續研究中以 250 天的模擬資料進行 CNN 訓練。另外，本文研究的是單量多載量小車調度，後續研究將對多輛多載量小車協同調度、多物流區域聯合調度進行探討。

參考文獻

- [1] Chang, Q., Pan, C., Xiao, G., & Biller, S. (2013). Integrated modeling of automotive assembly line with material handling. *Journal of Manufacturing Science and Engineering*, 135(1), 011018.
- [2] Egbelu, P. J., & Tanchoco, J. M. (1984). Characterization of automatic guided vehicle dispatching rules. *The International Journal of Production Research*, 22(3), 359-374.
- [3] Sabuncuoglu, I. (1998). A study of scheduling rules of flexible manufacturing systems: a simulation approach. *International Journal of Production Research*, 36(2), 527-546.
- [4] Wu, L. H., Mok, P. Y., & Zhang, J. (2011). An adaptive multi-parameter based dispatching strategy for single-loop interbay material handling systems. *Computers in Industry*, 62(2), 175-186.
- [5] Hsieh, C. H., Cho, C., Yang, T., & Chang, T. J. (2012). Simulation study for a proposed segmented automated material handling system design for 300-mm semiconductor fabs. *Simulation Modelling Practice and Theory*, 29, 18-31.
- [6] Godinho Filho, M., Barco, C. F., & Neto, R. F. T. (2014). Using Genetic Algorithms to solve scheduling problems on flexible manufacturing systems (FMS): a literature survey, classification and analysis. *Flexible Services and Manufacturing Journal*, 26(3), 408-431.
- [7] Chen, C., Xi, L. F., Zhou, B. H., & Zhou, S. S. (2011). A multiple-criteria real-time scheduling approach for multiple-load carriers subject to LIFO loading constraints. *International Journal of Production Research*, 49(16), 4787-4806.
- [8] Fathi, M., Alvarez, M. J., Hassani Mehraban, F., & Rodríguez, V. (2014). A multiobjective optimization algorithm to solve the part feeding problem in mixed-model assembly lines. *Mathematical Problems in Engineering*, 2014.