کمینه‌سازی مدت زمان در جریان کار در محیط تولید کارگاه در جریان جایگشتی

امیر راست‌پور
دانشجوی کارشناسی ارشد مهندسی صنایع دانشگاه صنعتی اصفهان
arastpoor@yahoo.com
قاسم مصلحی
استادیار دانشکده مهندسی صنایع دانشگاه صنعتی اصفهان
moslehi@cc.iut.ac.ir

واژه‌های کلیدی
کارگاه ، توالی، ابتکاری، مدت زمان در جریان کار

چکیده
در این مقاله یک الگوریتم ابتکاری برای حل مسائل کمینه سازی میانگین مدت زمان در جریان کار در محیط تولید کارگاه در جریان جایگشتی ارائه شده است که این مسئله به شکل n/m/p/ $\overline{F}$ نمایش داده می‌شود. الگوریتم پیشنهادی برمبنای بهینه سازی توالی جزئی کارها استوار است که این روش به الگوریتم NEH [1] بر می‌گردد. کارایی الگوریتم پیشنهادی با الگوریتمهای شناخته شده‌ای مثل الگوریتم "فرامینان" و "لیسن"[2] و همچنینWY [3] مقایسه شده است. الگوریتم ارائه شده دارای سطح پیچیدگی کمتری نسبت به سایر الگوریتمها می‌باشد. جهت سنجش کارایی الگوریتم مسائلی با ۵ تا ۵۰ کار و ۵ تا ۲۰ ماشین تولید شده و سپس حل گردیده‌اند. نتایج بدست آمده نشان می‌دهند که الگوریتم ارائه شده از کارایی بهتری نسبت به سایر الگوریتمها برخوردار است.

مقدمه

سیستم تولید کارگاه در جریان جایگشتی یکی از سیستمهای متعارف تولید است. یک سیستم تولید کارگاهی توالیهای بسیار متنوعی را می‌تواند داشته باشد. این توالیها از روی معیارهای سنجش قابل دسته‌بندی هستند. هر توالی که بتواند این معیارها را به نحو بهتری ارضا نماید دارای اولویت بالاتری برای پیاده‌سازی خواهد بود. یکی از این معیارها، میانگین مدت زمان در جریان کار[1] است. این معیار به عنوان سنجه‌ای برای سنجش سرعت جریان کار، استفاده مناسب ازمنابع و محصول نیمه ساخته می‌باشد.

مسئله $n/m/p/\overline{F}$ به عنوان یک مساله NP-complete شناخته می‌شود. فلذا تحقیقات بسیاری برای توسعه یک الگوریتم ابتکاری که بتواند جوابی خوب (نه لزوما بهینه) را در زمان مناسب ارائه دهد، صورت می‌پذیرد.

در این مقاله سعی شده است روشی سریع وعملی جهت تعیین توالی کارها در محیط تولید کارگاه در جریان جایگشتی ارائه شود. روش پیشنهادی بر مبنای بهینه سازی توالی جزئی کارها استوار است که به الگوریتم NEH [1] بر می‌گردد.

مروری بر مطالعات گذشته

الگوریتمهای ابتکاری ارائه شده برای مسئله $n/m/p/\overline{F}$ بسیار متنوعند، از آن جمله می‌توان به "گوپتا" [4]، "راجندران" و "گودهوری" [5]، "راجندران" [6] ، "میازاکی" و همکاران[7]، "فرامینان" و "لیسن" [2] و "وو" و "ییم" [3] اشاره نمود.

"راجندران" و "گودهوری" سه الگوریتم ابتکاری توسعه و نشان دادند که الگوریتمهای ارائه شده از لحاظ کیفیت جواب و زمان عملکرد بهتر از الگوریتمهای "گوپتا" و "میازاکی" و همکاران عمل می‌کنند. "راجندران" یک الگوریتم جدید ارائه داده و نشان داد که این الگوریتم جدید از الگوریتم "راجندران" و "گودهوری" کارایی بالاتری دارد و در زمان کمتر جوابهای بهتری می‌دهد.

"وو" و "ییم" الگوریتم ابتکاری WY را برای مساله $n/m/p/\overline{F}$ ارائه دادند. آنها عملکرد الگوریتم خود را با الگوریتم ابتکاری "راجندران" مقایسه کردند. همچنین زمان ساخت[2] الگوریتم خود را با جواب الگوریتم هایی مثل NEH [1] و CDS[8] مقایسه و نشان دادند که الگوریتم آنها برتر از الگوریتم های معیار است.

در ادامه این مقاله مطالب با ترتیب زیر ارائه می‌شوند: در قسمت سوم الگوریتم "فرامینان" و "لیسن" [2] را ارئه می‌دهیم. در بخش چهارم یک الگوریتم ابتکاری جدید ارائه می‌دهیم که بر مبنای بهینه سازی زمان در جریان کار توالیهای جزئی استوار است. در بخش پنجم سطح پیچیدگی الگوریتم را محاسبه می‌کنیم. در قسمت ششم عملکرد الگوریتم پیشنهادی را با عملکرد الگوریتم های WY و "فرامینان" و "لیسن" مقایسه می‌کنیم. در نهایت در بخش هفتم نتایج بدست آمده را ارائه می‌کنیم.

---

1. Flowtime
2. Makespan

الگوریتم ابتکاری "فرامینان" و "لیسن"

گام ۱: قرار دهید K=1

- به ازای تمام کارها، زمان در جریان کار را محاسبه کنید. از آنجا که هیچ کاری قبل از آنها قرار نمی‌گیرد فلذا زمان در جریان با جمع زمانهای پروسس روی تمام ماشینها برابر می‌شود.
- کار، با کمترین زمان در جریان کار را از بین n کار انتخاب می‌کنیم. مجموعه S را به عنوان توالی جزئی تشکیل دهید.
- n-1 کار باقیمانده را در مجموعه عناصر باقیمانده(R) قرار دهید.

گام دوم: K=K+1 را بروز رسانی کنید.

گام سوم: کار J، $J \in R$ را در K جایگاه ممکن از جواب فعلی قرار دهید.

- از بین K توالی ایجاد شده بهترین را با کمترین میزان زمان درجریان کار انتخاب، و تحت نام C به عنوان جواب کاندیدا معرفی کنید.
- قرار دهید S:=C
- کار J را از مجموعه کارهای باقیمانده R حذف نمایید.

گام چهارم: اگر K>2، سپس به ازای کلیه $(1 \leq i < K, i < j < K)$ i, j تمام جابجایی های ممکن حاصل از تعویض جای کارهای i وj در مجموعه C را بدست آورید. اگر از بین تمام K.(K-1)/2 توالی جزئی حاصل یک توالی C' وجود داشته باشد که زمان در جریان کار آن کمتر ازS باشد در این صورت $S:=C'$.

گام پنجم: اگر مجموعه کارهای باقیمانده تهی باشد ویا بطور معادل K=n باشد پس به پایان الگوریتم رسیده‌اید. در غیر اینصورت به گام دوم بروید.

الگوریتم ابتکاری"فرامینان" و "لیسن" دارای سطح پیچیدگی$O(n^4 m)$ است[2].

الگوریتم ابتکاری پیشنهادی

الگوریتم ابتکاری پیشنهادی بر مبنای ایده الگوریتم ابتکاری NEH استوار است. الگوریتم ابتکاری NEH، ابتدا با چیدن نزولی کارها بر اساس جمع زمان‌های پروسس آنها یک جواب اولیه بدست می‌آورد. سپس کارها را یک‌به‌یک از این جواب اولیه برداشته و هر کار را در تمام جایگاههای ممکن توالی جزئی قرار می‌دهد و بهترین حالت را بدست می‌آورد. این الگوریتم نتایج بسیار عالی را از خود نشان داده است. می‌توان نشان داد که توانایی بالای این نحوه عملکرد بدلیل تعداد حالتهای مختلفی است که در طول اجرای الگوریتم چک می‌شود نه توالی اولیه آن[9]. نشان داده شده است که این الگوریتم بهترین نیست[11 3 10]. ما نیز از ایده اصلی این الگوریتم در تدوین الگوریتم ابتکاری خود استفاده کرده‌ایم. در ادامه، گامهای الگوریتم پشنهادی را ارائه می‌کنیم:

گام ۱: قرار دهید K=1

- به ازای تمام کارها، زمان در جریان کار را محاسبه کنید. از آنجا که هیچ کاری قبل از آنها قرار نمی‌گیرد فلذا زمان در جریان با جمع زمانهای پروسس روی تمام ماشینها برابر می‌شود.
- کار، با کمترین زمان در جریان کار را از بین n کار انتخاب می‌کنیم. مجموعه S را به عنوان توالی جزئی تشکیل دهید.
- n-1 کار باقیمانده را در مجموعه عناصر باقیمانده(R) قرار دهید.

گام دوم: K=K+1 را بروز رسانی کنید.

گام سوم: کار J، $J\in R$ را در K جایگاه ممکن از جواب فعلی قرار دهید.

از بین K توالی ایجاد شده بهترین را با کمترین میزان زمان کار درجریان انتخاب، و تحت نام C به عنوان جواب کاندیدا معرفی کنید.

- قرار دهید S:=C

کار J را از مجموعه کارهای باقیمانده R حذف نمایید.

گام چهارم: کلیه توالی‌های ممکن حاصل از تعویض ۶-۲ کار آخر توالی C را بدست آورید. اگر از بین تمام ۸۷۲ حالت ممکن (در بیشترین حالت ممکن)، توالی جزئی حاصل یک توالی C' وجود داشته باشد که زمان در جریان کار آن کمتر ازS باشد در این صورت $S:=C'$.

گام پنجم: اگر مجموعه کارهای باقیمانده تهی باشد ویا بطور معادل K=n باشد پس به پایان الگوریتم رسیده‌اید. در غیر اینصورت به گام دوم بروید.

محاسبه سطح پیچیدگی الگوریتم

گام سوم شامل قرار دادن کارJ در تمام K جایگاه ممکن می‌شود. به ازای هر جاگذاری زمان در جریان کار یکبار محاسبه می‌شود.

محاسبه زمان در جریان کار برای یک مساله $n/m/p/\overline{F}$ به شکل زیر صورت می‌پذیرد:

$$C_{i0}=0,\ C_{0j}=0,$$
$$C_{ij}=\max\{C_{i,j-1}, C_{i-1,j}\}+t_{ij}$$
$$(i=1,\ldots,n)\ (j=1,\ldots,m)$$

که در آن C_{ij} زمان تکمیل کار i روی ماشین j می‌باشد. پس زمان درجریان کار از رابطه زیر بدست می‌آید.

$$F=\sum_{i=1}^{n} C_{im}$$

از روی رابطه بالا کاملا واضح است که سطح پیچیدگی برای محاسبه مدت زمان در جریان کار برای یک مساله با n کار و m ماشین O(nm) می‌باشد. اگر الگوریتم را در مورد مساله‌ای با n کار و m ماشین اجرا کنیم بایستی در مرحله ۳ الگوریتم، nبار زمان کار در

جریان محاسبه شود فلذا مرحله ۳ الگوریتم دارای سطح پیچیدگی $O(n^2m)$ می‌باشد. از طرف دیگر مرحله سوم در یک حلقه n تایی اجرا می‌شود در نتیجه الگوریتم پیشنهادی ما دارای سطح پیچیدگی $O(n^3 m)$ می‌باشد.

سنجش عملکرد الگوریتم پیشنهادی

برای مقایسه کارایی الگوریتم پیشنهادی، دو الگوریتم WY و "فرامینان" و "لیسن" به عنوان الگوریتمهای معیار انتخاب شده‌اند چرا که این دو الگوریتم از جمله بهترین الگوریتمهای ابتکاری شناخته شده هستند[2]. برای سنجش کارایی الگوریتم پیشنهادی، مسائلی طبق ساختار طرح شده توسط "وو" و "ییم" [3] ایجاد شده‌اند. بر اساس این ساختار ابتداً دو دسته مساله طراحی و تولید می‌شوند. دسته اول شامل مسائل کوچک می‌شود که جواب بهینه آنها از طریق شمارش کامل قابل دستیابی است و دسته دوم، مسائل بزرگ که بدست آوردن جواب بهینه آنها زمان بسیار زیادی نیاز دارد. مسائل با تعداد ۸–۵ کار و ۵، ۱۰، ۱۵و ۲۰ ماشین، جزء مسائل دسته اول و مسائل با تعداد ۹، ۱۰، ۲۰، ۳۰، ۴۰، ۵۰ کار و ۵، ۱۰، ۱۵و ۲۰ ماشین، جزء مسائل دسته دوم قرار می‌گیرند.

برای هر مساله‌ای زمان پردازش کارها روی هر ماشین بطور تصادفی و با توزیع یکنواخت بین یک تا صد تولید شده است. به ازای هر ترکیبی از تعداد کارها وماشینها ۱۰۰ مساله تولید و با هر سه الگوریتم حل شده است پس بطور کلی ۴۰۰۰ مساله مختلف حل شده است.

بر اساس عملکرد "وو" و "ییم" دو معیار " میانگین نسبی درصد انحراف از بهترین جواب"[1] و تعداد "بهترین جواب" حاصل از هر الگوریتم، به عنوان دو معیار سنجش کارایی الگوریتم‌ها مورد استفاده قرار گرفته‌اند. معیار میانگین نسبی درصد انحراف به شکل زیر تعریف می‌شود:

$$\sum_{i=1}^{100}\left(\frac{huristic_i - bestknown_i}{bestknown_i}\right)/100$$

که در آن $best\ known_i$ همان "بهترین جواب" است. برای مسائل کوچک $best\ known_i$ عبارت است از جواب روش شمارش کامل و برای مسائل بزرگ کمترین زمان در جریان کار است که از الگوریتمها بدست آمده است.

نتایج بدست آمده از حل مسائل کوچک توسط الگوریتمهای مختلف در جدول شماره ۱ نشان داده شده است. در جدول شماره ۱ به ازای هر الگوریتم، دو ستون مجزا تحت عناوین"ARPD" و"تعداد جواب بهینه" وجود دارد که در اولی ARPD مربوطه با توجه به رابطه تعریف شده، محاسبه و جایگذاری شده است و در ستون دوم تعداد مسائلی که جواب بدست آمده برای آنها توسط الگوریتم ابتکاری مربوطه همان جواب بهینه حاصل از روش شمارش کامل باشد قید شده است. نمودارهای شماره ۱و ۲ نحوه عملکرد الگوریتمهای مختلف روی مسائل کوچک را نشان می‌دهند.

جدول شماره ۲ نشان دهنده نحوه عملکرد هر یک از الگوریتمها به ازای مسائل بزرگ می‌باشد. در این جدول به ازای هر الگوریتم، دو ستون مجزا تحت عناوین"ARPD" و"تعداد بهترین جواب" وجود دارد که در اولی ARPD مربوطه با توجه به رابطه تعریف شده، محاسبه و جایگذاری شده است و در ستون دوم تعداد مسائلی که جواب بدست آمده برای آنها توسط الگوریتم ابتکاری مربوطه بهترین

1. Average Relative Percentage Deviation (ARPD)

جواب در بین جوابهای حاصل از سه الگوریتم ابتکاری باشد، قید شده است. نمودارهای شماره۳و ۴ نحوه عملکرد الگوریتمهای مختلف روی مسائل بزرگ را نشان می‌دهند.

نمودارهای ۱و ۲ نمایش می‌دهند که برای مسائل کوچک، الگوریتم پیشنهادی هم از لحاظ تعداد جوابهای بهینه و هم از لحاظ درصد انحراف از جواب بهینه در حد قابل توجهی بهتر از الگوریتمهای دیگر خصوصا الگوریتم ابتکاری WY عمل می‌کند.

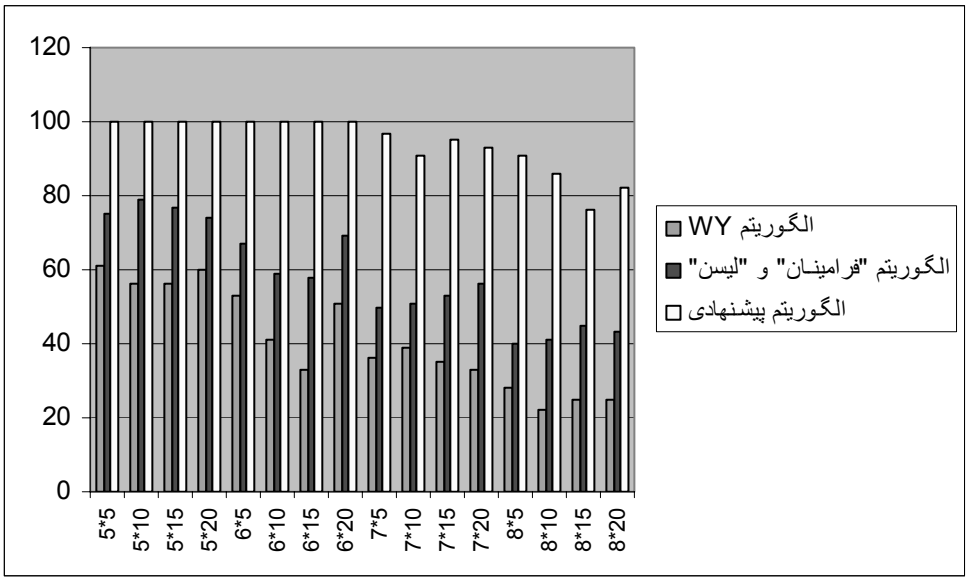
نمودارهای شماره ۳و ۴ نشان می‌دهند که الگوریتم پیشنهادی در مسائل بزرگ نیز عملکرد خوبی از خود نشان می‌دهد ولی به نظر می‌رسد که این کارایی متناسب با افزایش ابعاد مسئله کاهش پیدا می‌کند.

بحث و نتیجه‌گیری

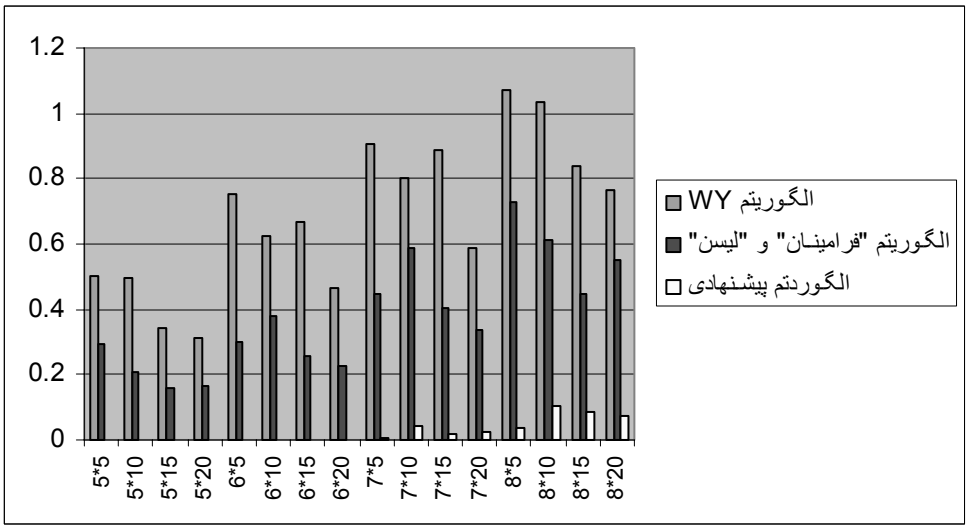
در این مقاله یک الگوریتم ابتکاری برای حل مسائل کمینه‌سازی متوسط مدت زمان در جریان کار($n/m/p/\overline{F}$) ارائه شد. این الگوریتم ابتکاری باوجود داشتن سطح پایین‌تری از پیچیدگی نسبت به الگوریتمهای WY و "فرامینان" و "لیسن"، بهتر از آنها عمل می‌کند. به دلیل کارایی خوب الگوریتم ارائه شده می‌توان آن را برای حل مسائل دیگر، مثل کمینه کردن ماکزیمم مدت زمان در جریان ($n/m/p/c_{max}$) نیز به کار برده و کارایی آن را بدست آورد. همچنین مناسب است که کارایی الگوریتم نسبت به ابعاد بزرگتر ماشین و کار سنجیده شود.

جدول۱: جوابهای سه الگوریتم به ازای مسائل کوچک.

		الگوریتم WY		"فرامینان" و "لیسن"		الگوریتم پیشنهادی	
n	m	ARPD%	تعداد جواب بهینه	ARPD%	تعداد جواب بهینه	ARPD%	تعداد جواب بهینه
۵	۵	**0.50**	**61**	**0.29**	**75**	**0**	**100**
	۱۰	**0.49**	**56**	**0.20**	**79**	**0**	**100**
	۱۵	**0.34**	**56**	**0.16**	**77**	**0**	**100**
	۲۰	**0.31**	**60**	**0.16**	**74**	**0**	**100**
۶	۵	**0.75**	**53**	**0.30**	**67**	**0**	**100**
	۱۰	**0.62**	**41**	**0.37**	**59**	**0**	**100**
	۱۵	**0.67**	**33**	**0.25**	**58**	**0**	**100**
	۲۰	**0.46**	**51**	**0.22**	**69**	**0**	**100**
۷	۵	**0.90**	**36**	**0.44**	**50**	**0**	**97**
	۱۰	**0.80**	**39**	**0.58**	**51**	**0.04**	**91**
	۱۵	**0.88**	**35**	**0.40**	**53**	**0.01**	**95**
	۲۰	**0.59**	**33**	**0.33**	**56**	**0.02**	**93**
۸	۵	**1.07**	**28**	**0.73**	**40**	**0.03**	**91**
	۱۰	**1.03**	**22**	**0.61**	**41**	**0.10**	**86**
	۱۵	**0.84**	**25**	**0.44**	**45**	**0.083**	**76**
	۲۰	**0.76**	**25**	**0.54**	**43**	**0.075**	**82**



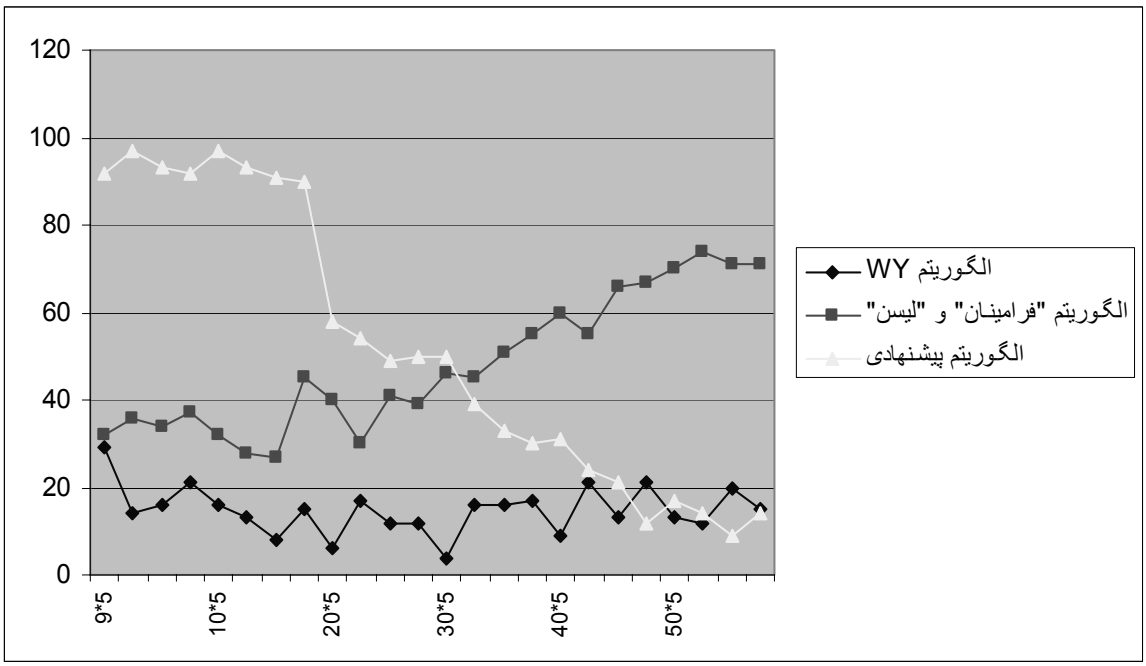
نمودار ۱: تعداد جوابهای بهینه سه الگوریتم در مسائل کوچک.



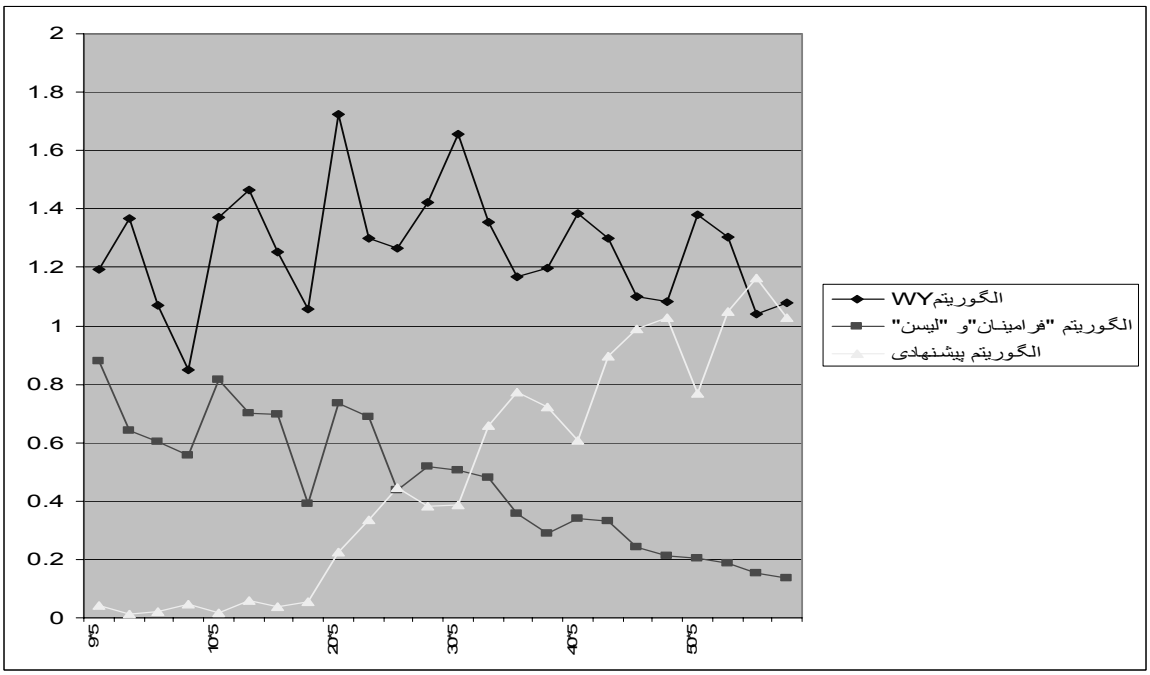
نمودار ۲: ARPD سه الگوریتم در مسائل کوچک.

جدول ۲: جوابهای سه الگوریتم برای مسائل بزرگ

		الگوریتم WY		"فرامینان" و "لیسن"		الگوریتم پیشنهادی	
n	m	ARPD%	تعداد بهترین جواب	ARPD%	تعداد بهترین جواب	ARPD%	تعداد بهترین جواب
۹	۵	**1.19**	**29**	**0.87**	**32**	**0.04**	**92**
	۱۰	**1.36**	**14**	**0.64**	**36**	**0.01**	**97**
	۱۵	**1.07**	**16**	**0.60**	**34**	**0.01**	**93**
	۲۰	**0.84**	**21**	**0.55**	**37**	**0.04**	**92**
۱۰	۵	**1.37**	**16**	**0.81**	**32**	**0.01**	**97**
	۱۰	**1.46**	**13**	**0.70**	**28**	**0.05**	**93**
	۱۵	**1.25**	**8**	**0.69**	**27**	**0.03**	**91**
	۲۰	**1.05**	**15**	**0.38**	**45**	**0.05**	**90**
۲۰	۵	**1.72**	**6**	**0.73**	**40**	**0.22**	**58**
	۱۰	**1.29**	**17**	**0.68**	**30**	**0.33**	**54**
	۱۵	**1.26**	**12**	**0.43**	**41**	**0.44**	**49**
	۲۰	**1.42**	**12**	**0.51**	**39**	**0.38**	**50**
۳۰	۵	**1.65**	**4**	**0.50**	**46**	**0.38**	**50**
	۱۰	**1.35**	**16**	**0.47**	**45**	**0.65**	**39**
	۱۵	**1.16**	**16**	**0.35**	**51**	**0.77**	**33**
	۲۰	**1.19**	**17**	**0.28**	**55**	**0.72**	**30**
۴۰	۵	**1.38**	**9**	**0.34**	**60**	**0.60**	**31**
	۱۰	**1.29**	**21**	**0.33**	**55**	**0.89**	**24**
	۱۵	**1.09**	**13**	**0.24**	**66**	**0.98**	**21**
	۲۰	**1.08**	**21**	**0.21**	**67**	**1.02**	**12**
۵۰	۵	**1.38**	**13**	**0.20**	**70**	**0.76**	**17**
	۱۰	**1.30**	**12**	**0.18**	**74**	**1.04**	**14**
	۱۵	**1.03**	**20**	**0.15**	**71**	**1.16**	**9**
	۲۰	**1.07**	**15**	**0.13**	**71**	**1.02**	**14**



نمودار ۳: تعداد "بهترین جواب" سه الگوریتم در مسائل بزرگ.



نمودار۴: ARPD سه الگوریتم در مسائل بزرگ.

منابع و مراجع

1. Nawas M., Enscore E.E., Ham I., "A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem", Omega, 11, 1983, 91-5.
2. J.M. Framinan, R. Leisten, "An efficient constructive heuristic for flowtime minimization in permutation flow shops", Omega, 31, 2003, 311-17
3. Woo D.S., Yim H.S., "A heuristic algorithm for mean flowtime objective in flowshop scheduling", Computers and Operations Research, 25, 1998, 175-82
4. Gupta JND., "Heuristic algorithms for multistage flowshop scheduling problem", AIIE Transactions, 4, 1972, 11-8
5. Rajendran C., Chaudhuri D., "An efficient heuristic approach to the scheduling of jobs in a flowshop", European Journal of Operational Research, 61, 1991, 318-25
6. Rajendran C., "Heuristic algorithm for scheduling in a flowshop to minimize total flowtime", International journal of production Economics, 29, 1993, 65-73.
7. Miyazaki S., Nishiyama N., Hashimoto F., "An adjacent pairwise approach to the mean flowtime scheduling problem", Journal of Operations Research society of Japan, 21, 1978, 287-99
8. Campbell H.G., Dudek R.A., Smith ML., " A heuristic algorithm for the n-job, m-machine sequencing problem", Management science, 16, 1970, B630-7
9. Framinan J.M., Leisten R., Ruiz-Usano R., "Efficient heuristics for flowshop sequencing with the objectives of makespan and flowtime minimization", European Journal of Operational Research, 141, 2002, 561-71
10. Allahverdi A., Aldowaisan T., "New heuristic to minimize total completion time in m-machine flowshops", International Journal of Production Economics, 77, 2002, 71-83
11. Framinan J.M., Leisten R., Rajendran C., "Different initial sequences for the heuristic of Nawaz, Enscore and Ham to minimize makespan, idletime or flowtime in the static permutation flowshop sequencing problem", International Journal of Production Research, 41, 2003, 121-48.