

ارائه یک الگوریتم ابتکاری برای مسئله زمان بندی پروژه با هدف حداکثر کردن خالص ارزش فعلی

قاسم مصلحی* و هادی قهار**

دانشکده مهندسی صنایع و سیستمها، دانشگاه صنعتی اصفهان

(دریافت مقاله: ۸۳/۱۰/۳ - دریافت نسخه نهایی: ۸۵/۳/۲۸)

چکیده - در این مقاله مسئله زمان بندی پروژه با هدف ماکزیمم کردن خالص ارزش فعلی بدون محدودیت منابع بررسی شده است. الگوریتمی به نام الگوریتم تفاضلی برای این منظور ارائه شده است. برای ارزیابی کارایی این الگوریتم شبکه هایی با تعداد ۱۰ تا ۱۰۰۰ گره و با ضریب پیچیدگی شبکه^۱ بین ۱/۳ تا ۶/۶ تولید شده است. مقدار خالص ارزش فعلی و مدت زمان حل مسائل در الگوریتم تفاضلی با مقدار خالص ارزش فعلی و مدت زمان جوابهای به دست آمده از الگوریتم جستجوی برگشتی مقایسه شده است. نتایج محاسباتی نشان می دهد که الگوریتم تفاضلی با توجه به پارامترهای تعداد گره، ضریب پیچیدگی شبکه و سررسید^۲ پروژه از کارایی خوبی برخوردار است. واژگان کلیدی: مسئله زمان بندی پروژه، خالص ارزش فعلی، روش تفاضلی، روش جستجوی برگشتی، جریان نقدی.

A Heuristic Algorithm for Project Scheduling Problem to Maximizing the Net Present Value

G. Moslehi and H. Ghahar

Department of Industrial Engineering and Systems, Isfahan University of Technology

Abstract: This paper deals with resource unconstrained project scheduling problems with the objective of maximizing the net present value (NPV) of project cash flows. Here we present a heuristic algorithm named as differential procedure (Dif_AOA). In order to evaluate the efficiency of this algorithm, networks with node numbers between 10-1000 and network complexity coefficients between 1.3-6.6 have been generated. We have compared both the total time for solving the problem and NPV of the Dif_AOA with those of the recursive search procedure. Computational results show that the Dif_AOA performs very effectively. Extensive analysis have been performed to evaluate the node number, complexity network coefficients (CNC), and deadline.

Keywords: Project scheduling problems, Net present value, Differential procedure, Recursive search procedure, Cash flow.

** - کارشناس ارشد

* - دانشیار

مسئله زمان بندی پروژه^۳ عبارت است از تعیین تاریخ انجام فعالیتهای یک پروژه برای رسیدن به یک هدف معین. این هدف می‌تواند تکمیل هر چه زودتر پروژه و یا حداکثر کردن منافع ناشی از اجرای پروژه باشد. اکثر متدلوژی‌هایی که در موضوع زمان بندی پروژه در مقالات معرفی شده‌اند به هدف حداقل کردن مدت زمان پروژه با انواع مختلف محدودیتهای منابع و محدودیتهای پیش‌نیازی پرداخته‌اند. لذا جنبه‌های مالی پروژه اغلب نادیده گرفته شده [۱]، و یا به عنوان معیار درجه دوم مدنظر است [۲]. هنگامی که جنبه‌های مالی مدیریت پروژه مدنظر باشد، تابع هدف حداکثر کردن خالص ارزش فعلی، یکی از مناسبترین معیارها به شمار می‌رود [۱]. در سال ۱۹۷۰ برای اولین بار مسئله زمان بندی پروژه با هدف حداکثر کردن خالص ارزش فعلی جریانهای نقدی پروژه (که از این به بعد به صورت PSP_MAX_NPV نمایش داده می‌شود)، توسط راسل ارائه شد [۳]. بعد از راسل، مقالات زیادی راجع به این مسئله نوشته شده است که هر کدام به گونه‌ای به بسط این مسئله در زوایای گوناگون پرداخته‌اند.

اهمیت استفاده از معیار خالص ارزش فعلی در زمان بندی پروژه در شرایط وجود محدودیت سرمایه، هزینه فرصت بالای سرمایه گذاری (نرخ بهره بالا)، مدت اجرای طولانی پروژه و نیز بالا بودن حجم جریانهای نقدی پروژه، بیشتر کرد پیدا می‌کند و هر چه این شرایط تشدید شود اهمیت استفاده از این معیار نمایانتر می‌شود.

مسئله PSP_MAX_NPV ، از ابتدای مطرح شدنش در سال ۱۹۷۰ تاکنون از جنبه‌های مختلفی بررسی شده است [۱]. همه کارهایی که تا کنون روی این مسئله صورت گرفته، می‌توان در چهار زمینه تقسیم کرد: ۱- اضافه کردن محدودیت منابع به مسئله. ۲- در نظر گرفتن حالت‌های متفاوت شبکه. ۳- در نظر گرفتن الگوهای مختلف جریانهای نقدی با توجه به واقعیت. ۴- ارائه الگوریتمهای مختلف برای حل مسئله.

برای حل PSP_MAX_NPV نامحدود، تاکنون چند روش

حل در مقالات ارائه شده است که بعضی دقیق و بعضی ابتکاری بوده‌اند. اولین روش ارائه شده برای حل این مسئله توسط راسل مطرح شد. راسل مدل ریاضی این مسئله را که به صورت غیر خطی بود، از طریق جمله اول بسط سری تیلور تخمین زد و نشان داد که شکل ثانویه این مسئله، حالتی از مسئله شبکه جریانهاست و به این طریق مسئله را حل کرد [۱].

روش حل دقیق دیگری که برای این مسئله ارائه شد، روش جستجوی برگشتی بود که توسط وانهوک و همکاران ارائه شد. این روش پس از زمان بندی اولیه همه گره‌ها در زودترین زمانشان با استفاده از یک تابع برگشتی مجموعه گره‌هایی که قابلیت انتقال به جلو دارند را پیدا می‌کند و به اندازه‌ای که مجاز باشند انتقال می‌دهد [۴].

برای حل این مسئله روشهای ابتکاری نیز ارائه شده است که عبارت‌اند از روش درون یافتی مغربی و همکاران [۵]، روش شبیه سازی سرد شدن (SA) [۶] اتگار و همکاران [۶]، و روش شاخه و کران اشتاب و اتگار [۲]. اتگار و همکاران فرض جریانهای نقدی غیر فزاینده در مسئله - در حالت AOA - وارد کردند و برای حل آن نیز روش مغربی را توسعه داده و از آن استفاده کردند [۷]. نهایتاً در سال ۲۰۰۱ وانهوک و همکاران فرض جریان نقدی خطی غیر فزاینده را در حالت شبکه AON در نظر گرفتند [۸] و برای حل آن نیز روش جستجوی برگشتی مرجع [۴] را توسعه دادند [۸].

از بین دو روش شاخه و کران و SA که نویسندگان هر دو آنها مشترک بودند، بنابر نتایجی که خود آنها ارائه کرده بودند، روش شاخه و کران دارای کیفیت و کارایی به مراتب بهتری نسبت به روش SA است [۲]. اما از آنجایی که این الگوریتم برای الگوی جریان نقدی پله‌ای غیر فزاینده نوشته شده بود، و تعداد پله‌های جریان نقدی هر گره تأثیر زیادی در کارایی الگوریتم داشت، هنگامی که از این الگوریتم برای الگوی جریان نقدی خطی غیر فزاینده استفاده شد، به دلیل اینکه در این حالت تعداد پله‌ها، یعنی تعداد گزیننده‌های قابل محاسبه زیاد می‌شد (به خصوص هنگام دور شدن سررسید پروژه)، مدت زمان

محاسبات به طور فزاینده افزایش می‌یافت.

در این مقاله یک الگوریتم ابتکاری برای PSP_MAX_NPV نامحدود ارائه خواهد شد. این الگوریتم ضمن برخورداری از منطق ساده، حتی هنگام دور شدن سررسید پروژه، مدت زمان بسیار کمی برای محاسبات نیاز دارد. در قسمت بعد به توصیف مسئله و فرضیات آن پرداخته شده است. در قسمت سوم الگوریتم ابتکاری پیشنهادی (الگوریتم تفاضلی) به طور مشروح توضیح داده شده است. در قسمت چهارم نتایج محاسباتی به دست آمده از اجرای الگوریتم تفاضلی و مقایسه آن با الگوریتم جستجوی برگشتی آمده است. در نهایت در قسمت آخر نتیجه گیری و پیشنهادات وجود دارد.

۲- توصیف و فرضیات مسئله

مسئله زمان بندی پروژه با معیار حداکثر کردن خالص ارزش فعلی نامحدود، عبارت است از، زمان بندی فعالیت‌های یک پروژه درون یک محدوده زمانی، در غیاب محدودیت منابع، به گونه‌ای که خالص ارزش فعلی جریانهای نقدی کل پروژه حداکثر شود. این محدوده زمانی، که خالص ارزش فعلی پروژه در درون آن باید حداکثر شود، حداقل به اندازه مسیر بحرانی شبکه و حداکثر به اندازه سررسید پروژه است.

مسئله زمان بندی پروژه از دو دیدگاه قابل بررسی است. یک دیدگاه، نقطه نظر کارفرما و دیدگاه دیگر، نقطه نظر پیمانکار است. در این تحقیق مسئله زمان بندی پروژه با هدف حداکثر کردن خالص ارزش فعلی از دیدگاه پیمانکار بررسی می‌شود. لذا جریان نقدی مثبت نشان دهنده دریافت‌های پیمانکار و جریان نقدی منفی نشان دهنده پرداخت‌های پیمانکار به پیمانکاران فرعی است.

در کارهایی که تاکنون روی PSP_MAX_NPV صورت گرفته است، فرض شده است که جریانهای نقدی یا رویدادگرا و یا فعالیت گرا هستند. در حالت فعالیت گرا به ازای هر فعالیت یک جریان نقدی خالص وجود دارد که عبارت از

خالص دریافت‌های پیمانکار از کارفرما و پرداخت‌های او به پیمانکار فرعی است و در حالت رویدادگرا مجموعه‌ای از یک یا چند فعالیت، دارای یک جریان نقدی خالص به صورت مشترک‌اند که از جمع خالص جریان نقدی هر کدام از این فعالیتها به دست آمده است. نکته‌ای که به نظر می‌رسد تاکنون به بررسی آن پرداخته نشده است حالت جریان نقدی توأم رویدادگرا و فعالیت گراست. یکی از مصادیق این حالت جریان نقدی در عالم خارج، حالتی است که کارفرما علاوه بر اینکه به ازای هر فعالیت یک جریان نقدی به پیمانکاری پردازد، به ازای تکمیل به موقع یک یا چند فعالیت (وقوع یک برهه^۱) نیز یک جریان نقدی مثل پاداش به پیمانکار پرداخت می‌شود.

حالت نمایش شبکه در این تحقیق AOA در نظر گرفته شده است. هر کمان نشان دهنده یک فعالیت است که دارای یک جریان نقدی مربوط به خود است. هرگره نشان دهنده تکمیل و یا شروع یک یا چند فعالیت است. جریان نقدی هر گره نشان دهنده خالص پاداش و جریمه‌هایی است که مربوط به تکمیل به موقع یا دیرکرد مجموعه فعالیت‌هایی است که به این گره وارد می‌شوند. روابط پیش نیازی مسئله از نوع پایان به شروع با زمان تأخیر صفر است. الگوی جریان نقدی مسئله نیز به صورت خطی غیر فزاینده در نظر گرفته شده است.

با توجه به فرضیات، فرمولبندی مسئله با توجه به جریان نقدی رویدادگرا و فعالیت گرا و نیز الگوی جریان نقدی خطی غیر فزاینده و حالت نمایش شبکه AOA به صورت معادلات (۱) تا (۵) است:

$$\text{Maximize: } \sum_{i=1}^{NN} C_i \beta^{t_i} + \sum_{i=1}^{NN} \sum_{j=1}^{NA} C_{ij} \beta^{t_{ij}} \quad (1)$$

$$\text{Subject To: } t_i + d_{ij} \leq t_j \quad \forall (i, j) \quad (2)$$

$$t_{ij} \geq t_i + d_{ij} \quad \forall (i, j) \quad (3)$$

$$t_{ij} \leq t_j \quad \forall (i, j) \quad (4)$$

$$t_{NN} \leq \delta \quad \forall (i, j) \quad (5)$$

که در آن:

$C_i =$ جریان نقدی گره i که به صورت $C_i = a + b_i$ (خطی) است.

$\beta =$ عامل تنزیل

$t_i =$ زمان تحقق گره i

$C_{ij} =$ جریان نقدی فعالیت $i-j$

$t_{ij} =$ زمان تکمیل فعالیت $i-j$

$NN =$ تعداد گره‌ها (رویدادها)

$NA =$ تعداد کمانها (فعالیتها)

$\delta =$ سررسید پروژه

معادله (۱)، خالص ارزش فعلی کل پروژه را در حالت رویدادگرا و فعالیت گرا حداکثر می‌کند. معادله (۲)، محدودیت پیش‌نیازی از نوع پایان به شروع با زمان تأخیر صفر را تأمین می‌کند. محدودیت (۳) تضمین می‌کند که تاریخ تکمیل فعالیت $i-j$ حداقل برابر تاریخ تحقق گره i به علاوه مدت زمان فعالیت $i-j$ باشد. قید (۴) بیان می‌کند که تاریخ تکمیل هر فعالیت نباید از تاریخ وقوع گره پایانی آن تجاوز کند و نهایتاً قید (۵) تصریح می‌کند که زمان بندی فعالیت‌های پروژه باید در محدوده سررسید پروژه صورت بگیرد.

همان گونه که در فرمولبندی مسئله مشاهده می‌شود، فرض جریان نقدی توأم روی گره‌ها و فعالیتها، تعداد متغیرهای مسئله را به تعداد کمانها افزایش می‌دهد و نیز تعداد محدودیت‌های مسئله به میزان قابل توجهی افزایش پیدا می‌کند. با استفاده از یک استدلال ساده می‌توان مسئله را ساده تر کرد. این استدلال در قالب یک مثال نشان داده خواهد شد.

اگر فعالیت $i-j$ در نظر گرفته شود که هر کدام از گره‌ها دارای یک جریان نقدی است و فعالیت میان آنها نیز به طور مستقل دارای یک جریان نقدی است. اگر فرض شود جریان نقدی گره i ، 500 و جریان نقدی گره j ، -700 و جریان نقدی فعالیت $i-j$ ، -1000 باشد، روشن است که برای حداکثر کردن خالص ارزش فعلی جریانهای نقدی پروژه، باید گره i در زودترین زمان ممکن و گره j در دیرترین زمان ممکن، زمان بندی شوند. فعالیت $i-j$ نیز که دارای جریان نقدی منفی است،

باید در دیرترین زمان ممکن زمان بندی شود.

همان طور که ملاحظه می‌شود به دلیل اینکه جریان نقدی فعالیت $i-j$ منفی است، زمان تکمیل آن مساوی با زمان وقوع گره پایانی است. به عبارت دیگر زمان تکمیل هر فعالیت حداقل برابر زمان وقوع گره ابتدایی آن به علاوه مدت زمان اجرای آن فعالیت است و حداکثر، برابر زمان گره پایانی آن است. به این ترتیب به جای اینکه به ازای هر فعالیت یک متغیر به مسئله اضافه شود، می‌توان جریان نقدی هر فعالیتی که دارای جریان نقدی منفی است به گره پایانی آن اضافه کرد و به طور مشابه جریان نقدی هر گره دارای جریان نقدی مثبت به میزان مدت آن تنزیل کرد و به جریان نقدی گره ابتدایی اضافه کرد. به این ترتیب، مسئله به یک مسئله ساده با جریان نقدی رویدادگرا تبدیل می‌شود، و فرمولبندی مسئله به صورت معادلات (۶) تا (۸) در می‌آید:

$$\text{Maximize : } \sum_{i=1}^{NN} C_i \beta^{t_i} \quad (6)$$

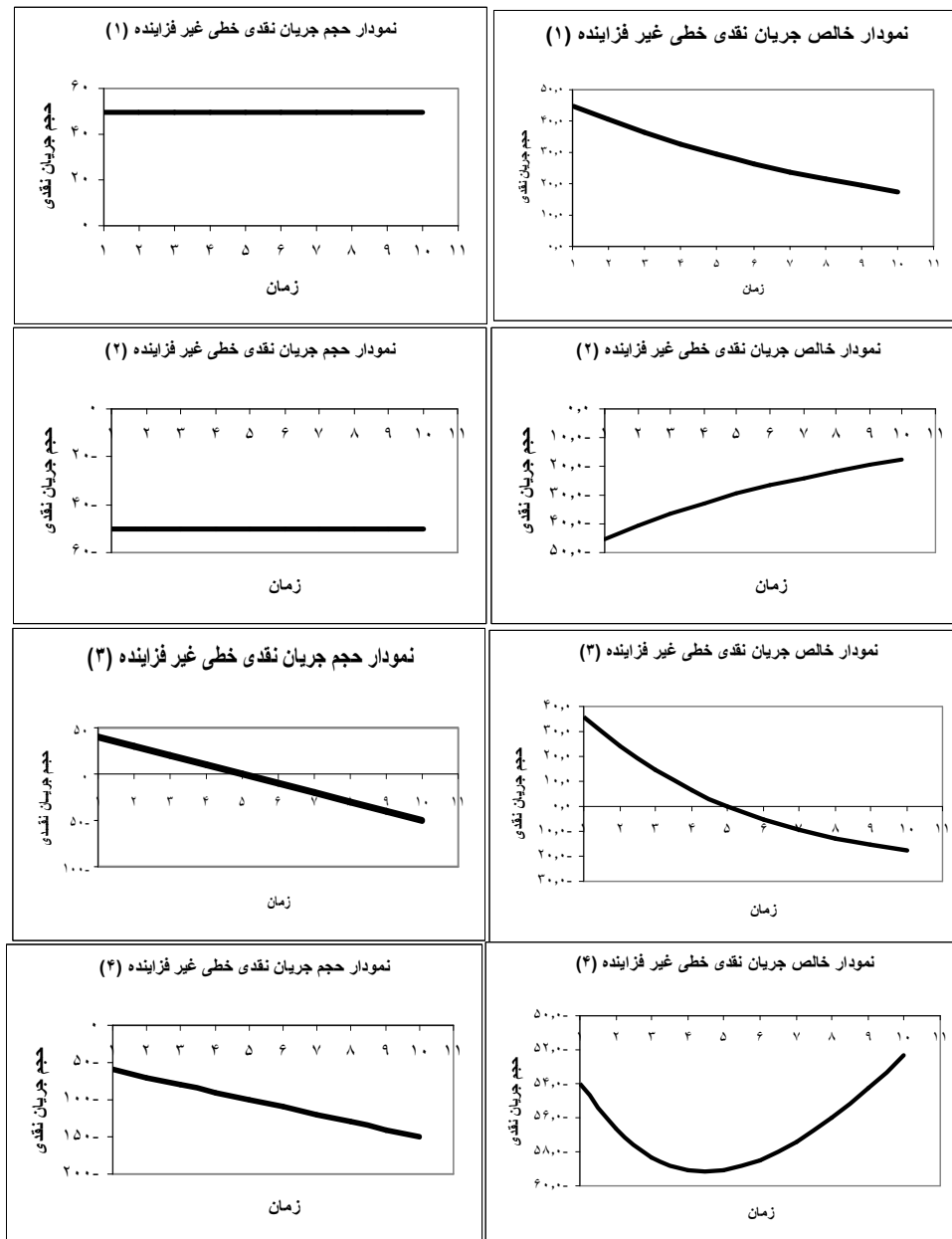
$$\text{Subject To : } t_i + d_{ij} \leq t_j \quad \forall (i, j) \quad (7)$$

$$t_{NN} \leq \delta \quad (8)$$

بعد از زمان بندی گره‌های مسئله می‌توان به راحتی تاریخ تکمیل فعالیتها را نیز به دست آورد. به این صورت که اگر جریان نقدی فعالیت مثبت است، تاریخ تکمیل آن عبارت است از تاریخ تحقق گره ابتدایی به علاوه مدت زمان اجرای آن فعالیت و اگر جریان نقدی فعالیت منفی است، تاریخ تکمیل آن فعالیت برابر با تاریخ وقوع گره پایانی است.

۳- الگوریتم تفاضلی (Dif_AOA)

الگوریتم ابتکاری ارائه شده، الگوریتم تفاضلی نامیده شده است. اساس الگوریتم تفاضلی، نکته‌ای است که در مقاله اتگار و اشتاب ارائه شده است. آنها این نکته را ذکر می‌کنند که با وجود جریان نقدی خطی غیر فزاینده، نقطه ماکزیمم خالص ارزش فعلی هر گره، الزاماً در زودترین زمان و یا دیرترین زمان ممکن تحقق آن است [۷]. شکل (۱) این نکته را به خوبی نشان



شکل ۱- چهار حالت الگوی جریان نقدی خطی غیر فزاینده و تابع خالص ارزش فعلی هر کدام

راست شکل (۱)، نمودار خالص ارزش فعلی این چهار حالت را نشان می‌دهد. همان گونه که در نمودارهای سمت راست شکل (۱) ملاحظه می‌شود، نقطه ماکزیمم خالص ارزش فعلی هر کدام از این چهار حالت یا در زودترین زمان و یا در دیرترین زمان آنهاست. با توجه به این نکته، هنگامی که در یک شبکه در ابتدا همه گره‌ها در زودترین زمان ممکن زمان بندی شده باشند، فقط گره‌هایی برای افزایش خالص ارزش

می‌دهد. در این شکل مشخص است هنگامی که جریان نقدی به صورت خطی غیر فزاینده در نظر گرفته شود، تابع حجم جریان نقدی ممکن است چهار صورت پیدا کند: (۱) حالت جریان نقدی ثابت مثبت ($a > 0, b = 0$)، (۲) حالت جریان نقدی ثابت منفی ($a < 0, b = 0$)، (۳) حالت جریان نقدی خطی با عرض از مبدأ مثبت ($a > 0, b < 0$)، (۴) حالت جریان نقدی خطی با عرض از مبدأ منفی ($a < 0, b < 0$). نمودارهای سمت

فعلی کل پروژه، قابلیت انتقال به جلو را دارند که خالص ارزش فعلی جریان نقدی آنها در دیرترین زمان بیشتر از خالص ارزش فعلی آنها در زودترین زمان باشد. به عبارت دیگر فقط هنگامی که تفاوت خالص ارزش فعلی یک گره در زمان فعلی و دیرترین زمان ممکن منفی باشد، امکان انتقال به جلوی آن بررسی می‌شود. بر این اساس، الگوریتم ابتکاری ارائه شده در این مقاله (الگوریتم تفاضلی)، در ابتدا همه گره‌های شبکه را در زودترین زمان ممکن زمان بندی می‌کند و خالص ارزش فعلی کل پروژه را محاسبه می‌کند و به عنوان بهترین جواب این مرحله ذخیره می‌کند. در الگوریتم تفاضلی در گام بعدی، گره‌ها را از آخر به اول بررسی می‌کند، هر گرهی که خالص ارزش فعلی آن در دیرترین زمان بیشتر از خالص ارزش فعلی در زمان فعلی باشد، به صورت موقت زمان خودش و گره‌های بعدی آن به سمت سررسید پروژه افزایش می‌یابد، اگر این افزایش باعث بهبود خالص ارزش فعلی کل شبکه شود، زمان گره‌ها و خالص ارزش فعلی کل پروژه ذخیره می‌شود و الگوریتم به سراغ گره‌های بعدی می‌رود. در غیر این صورت بدون ذخیره نتایج به سراغ گره بعدی می‌رود. در حین اجرای الگوریتم نکات دیگری نیز در نظر گرفته شده است که رعایت آنها برای تولید جوابهای مناسب، لازم است. در ادامه به ذکر این نکات پرداخته می‌شود.

۳-۱- میزان انتقال گره‌های بعدی

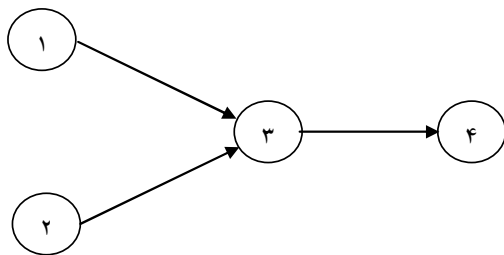
هنگامی که انتقال به جلوی یک گره مجاز تشخیص داده شد، قبل از اینکه آن گره انتقال یابد، باید تأثیر انتقال زمان این گره بر گره‌های بعدی آن نیز در نظر گرفته شود. برای این کار زمان گره تحت بررسی به اندازه دیرترین زمان آن و گره‌های بعدی آن به اندازه‌ای که مورد نیاز باشد انتقال داده می‌شوند. این اندازه مورد نیاز به این معنی است که ممکن است یکی یا چند تا از گره‌های بعدی گره تحت بررسی، قبلاً هنگام بررسی گره‌های دیگر انتقال پیدا کرده باشند و نیازی به انتقال نداشته

باشند و یا اینکه نیاز به انتقال کمتری داشته باشند. در این الگوریتم برای انتقال گره‌های بعدی از یک تابع برگشتی استفاده شده است. در این تابع اگر زمان گره‌های بعدی به گونه‌ای باشد که نیازی به تغییر زمان نداشته باشد (یعنی زمان گره i به علاوه مدت زمان فعالیت j ، کوچکتر از زمان گره j باشد)، روند انتقال از مسیر گره j متوقف می‌شود. اما اگر زمان گره‌های بعدی نیاز به تغییر داشته باشد (زمان گره i به علاوه مدت زمان فعالیت j ، بزرگتر از زمان گره j باشد)، آن‌گاه زمان گره بعدی (j)، به اندازه مورد نیاز $(T[i]+D[i-j]-T[j])$ ، انتقال پیدا می‌کند و تابع انتقال مجدداً با گره بعدی (j)، فراخوانی می‌شود.

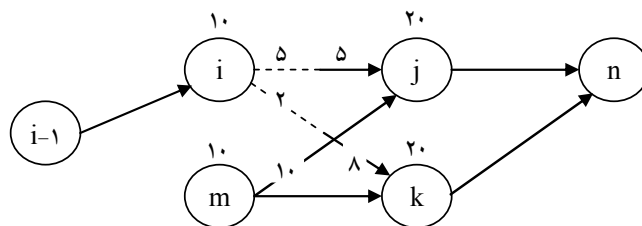
۳-۲- انتقال به اندازه حداقل فاصله با گره‌های بعدی

هنگام بررسی گره‌های شبکه برای امکان انتقال به جلو در درخت زودترین، ممکن است با گره‌هایی مواجه شویم که انتقال آنها از زمان فعلی به دیرترین زمان، به علت انتقال گره‌های بعدی به جلو، مقدار خالص ارزش فعلی شبکه را بهبود ندهد، اما انتقال آنها به اندازه حداقل فاصله گره با گره‌های بعدی آن خالص ارزش فعلی کل پروژه را افزایش دهد.

به عنوان مثال گره i دارای جریان نقدی منفی است، لذا انتقال آن به تنهایی باعث بهبود خالص ارزش فعلی پروژه می‌شود، اما به دلیل اینکه گره‌های بعدی آن دارای جریان نقدی مثبت‌اند، انتقال گره i به دیرترین زمان ممکن آن باعث کاهش خالص ارزش فعلی کل پروژه می‌شود، اما از طرف دیگر همان طور که در شکل (۲) ملاحظه می‌شود، فعالیت‌های بین این گره با گره‌های بعدی آن دارای ۵ و ۲ واحد شناوری‌اند. لذا امکان انتقال گره i به اندازه حداقل این دو شناوری وجود دارد و با توجه به منفی بودن جریان نقدی گره i ، انتقال این گره به اندازه حداقل شناوری فعالیت‌های بعدی آن باعث بهبود خالص ارزش فعلی کل پروژه می‌شود. محاسبه حداقل فاصله هر گره $(V(i))$ ، با گره‌های بعدی آن از طریق



شکل ۳- بررسی توام گره ها



شکل ۲- انتقال به اندازه حداقل فاصله با گره های بعدی

ممکن است این انتقال مفید باشد و خالص ارزش فعلی کل پروژه را افزایش دهد. یعنی رابطه زیر برقرار است.

$$NPV(1)_{Lr} + NPV(2)_{Lr} > NPV(3,4)_{shift} \quad (12)$$

برای در نظر گرفتن این نکته در الگوریتم از روش شمارش که در الگوریتم وانهوک و همکاران مورد استفاده قرار گرفته، استفاده می شود [۸]. به این صورت که بعد از بررسی تمام گره ها در مرحله قبلی، همه گره هایی که هنوز قابلیت انتقال را به تنهایی داشته باشند، شناسایی می شوند و با استفاده از تابع شمارش برگشتی وانهوک و همکاران، در هر مرحله امکان انتقال توام آنها بررسی می شود. برای روشن شدن نحوه استفاده از الگوریتم شمارش در الگوریتم Dif_AOA فرض کنید شبکه ای با ۱۰ گره وجود دارد که پس از زمان بندی همه گره ها در زودترین زمان ممکن، همه گره هایی که انتقال آنها باعث افزایش خالص ارزش فعلی آنها (خالص ارزش فعلی خود گره) می شود و نیز مانعی بر سر راه انتقال آنها وجود ندارد به جلو انتقال داده می شوند. به علاوه فرض کنید که ۳ گره ۲، ۴ و ۵ از لحاظ جریان نقدی خودشان قابلیت انتقال را دارند (یعنی انتقال آنها باعث افزایش خالص ارزش فعلی آنها می شود)، اما به جلو انتقال پیدا نکرده اند. زیرا گره های دیگری از شبکه که جریان نقدی مثبت دارند و جلوی این گره ها واقع شده اند، مانع انتقال این گره ها هستند. اگر این ۳ گره هیچ گونه گره بعدی مشترکی با یکدیگر نداشته باشند، انتقال توام آنها نیز هیچ تأثیری در خالص ارزش فعلی کل شبکه ندارد. اما اگر این ۳ گره حداقل یک گره بعدی مشترک داشته باشند، انتقال توام آنها ممکن است خالص ارزش فعلی کل شبکه را افزایش دهد. الگوریتم شمارش در الگوریتم تفاضلی به این

معادله (۹) صورت می گیرد:

$$V(i) = \min\{T(i) - D(i, j) - T(i)\} \quad (9)$$

$$j \in \text{Successors}(i)$$

۳-۳- انتقال توام ۲ یا چند گره

نکته دیگری که در الگوریتم در نظر گرفته شده این است که چون بررسی گره ها به صورت جداگانه صورت می گیرد، اثرات انتقال توام دو یا چند گره با هم روی خالص ارزش فعلی کل پروژه در نظر گرفته نمی شود. برای روشن شدن مسئله به شکل (۳)، توجه کنید.

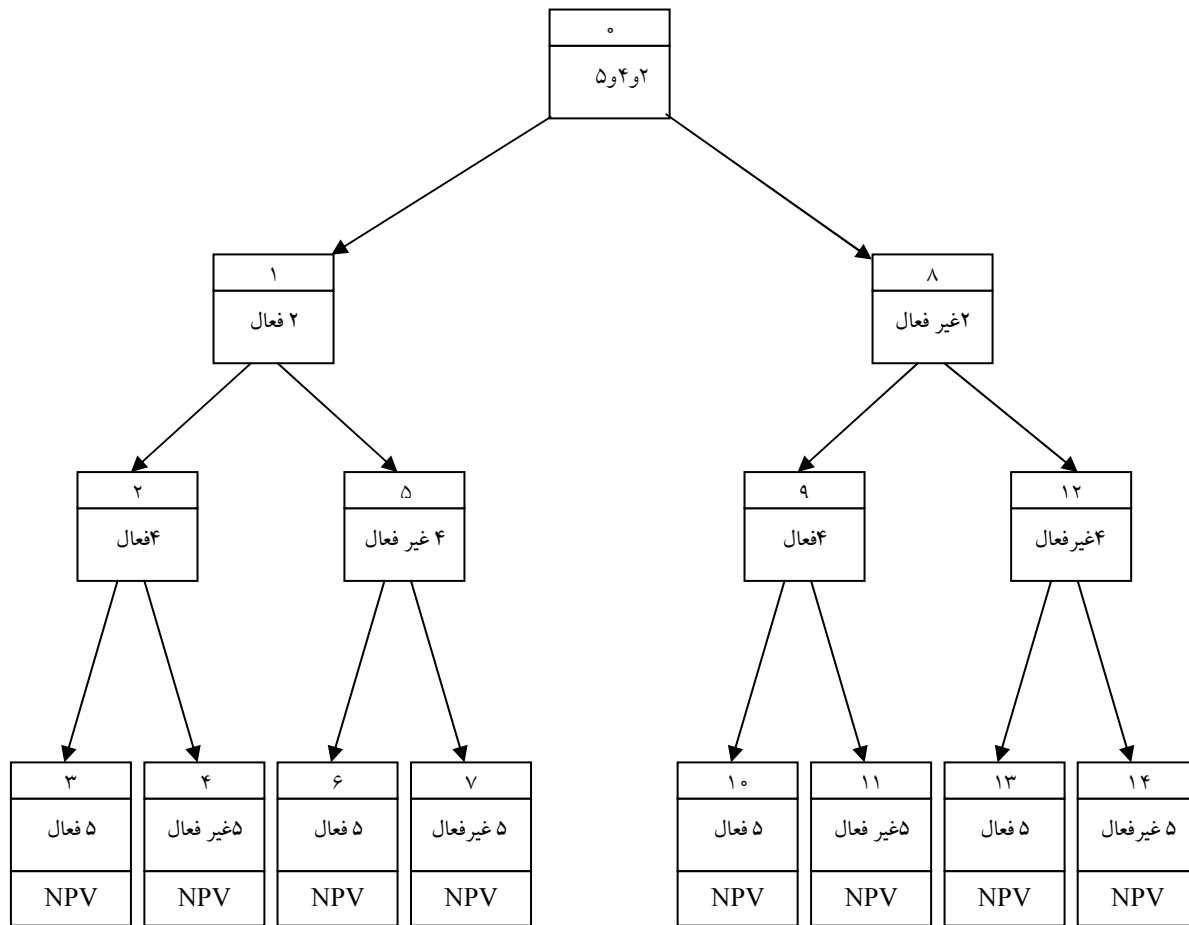
همان گونه که در شکل (۳) مشاهده می شود گره های ۱ و ۲ به گره ۳ منتهی می شوند. حال اگر فرض شود که گره های ۱ و ۲ هر دو دارای جریان نقدی باشند که به تأخیر افتادن آنها باعث افزایش خالص ارزش فعلی آنها می شود، و نیز اگر خالص ارزش فعلی گره ۳ و ۴ مثبت باشد، ممکن است حالتی پیش بیاید که هنگام بررسی گره های ۱ و ۲، حجم جریان نقدی هر کدام از این دو گره به میزانی باشد که نتوانند به تنهایی گره های ۳ و ۴ را به جلو انتقال دهند یعنی روابط زیر برقرار است.

$$NPV(1)_{Lr} < NPV(3,4)_{shift} \quad (10)$$

$$NPV(2)_{Lr} < NPV(3,4)_{shift} \quad (11)$$

رابطه (۱۰) و (۱۱) نشان می دهند که خالص ارزش فعلی گره های (۱) و (۲) به صورت جداگانه در دیرترین زمان ممکن (Lr) کمتر از خالص ارزش فعلی گره های ۳ و ۴ پس از انتقال^۱ است.

حال اگر انتقال گره های ۱ و ۲ به طور توام بررسی شود،

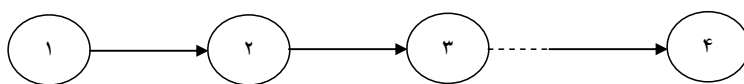


| |
|------------|
| شماره ردیف |
| |

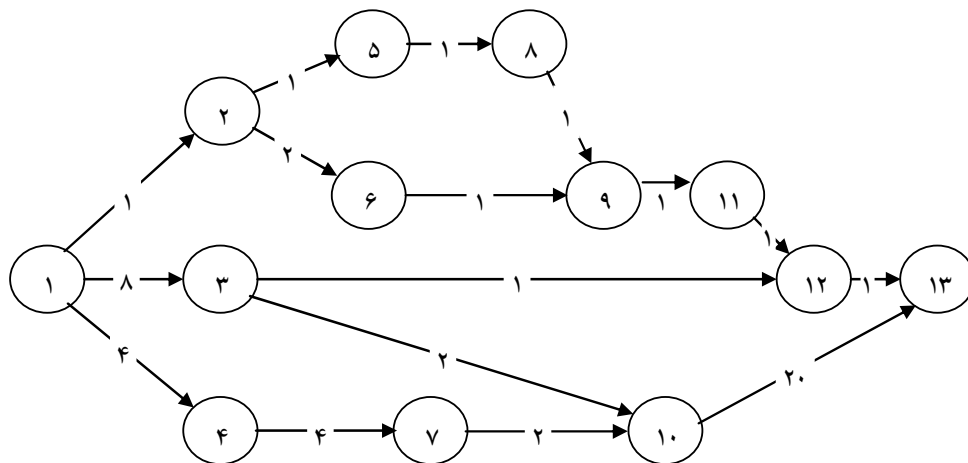
شکل ۴- نمودار الگوریتم شمارش برای انتقال توام

انتقال گره ۲ فعال می‌شود (جعبه ۱). الگوریتم شمارش پس از فعال کردن انتقال گره ۲ در سطح ۱، در سطح ۲ انتقال گره ۴ را فعال می‌کند (جعبه ۲)، به سطح ۳ می‌رود و انتقال گره ۵ را فعال می‌کند (جعبه ۳). در جعبه ۳ هر سه گره به صورت همزمان انتقال پیدا کرده‌اند. لذا در اینجا خالص ارزش فعلی کل شبکه محاسبه می‌شود. اگر خالص ارزش فعلی کل شبکه بهبود پیدا کرده بود، نتایج ذخیره می‌شود و اگر جواب بهبود پیدا نکرد، جواب ذخیره نمی‌شود و در همین سطح حالت غیرفعال انتقال گره ۵ بررسی می‌شود (جعبه ۴). سپس مجدداً خالص ارزش فعلی کل شبکه محاسبه می‌شود. اگر جواب بهبود یافته بود نتایج

صورت عمل می‌کند که پس از شناسایی همه گره‌هایی که به تنهایی قابلیت انتقال دارند اما منتقل نشده‌اند، در هر سطح، یکی از گره‌های مزبور را فعال می‌کند (یعنی خود گره و گره‌های بعدی آن را به جلو انتقال می‌دهد)، و به سطح بعد می‌رود و این روال تا جایی ادامه می‌یابد که همه حالات ممکن بین این ۳ گره بررسی شود. به این صورت انتقال توام این ۳ گره و نیز انتقال دو به دوی آنها بررسی می‌شود و در هر حالت اگر خالص ارزش فعلی کل پروژه بهبود یابد، نتایج ذخیره می‌شود. شکل (۴) روال کار الگوریتم شمارش را برای این مثال نشان می‌دهد. گره ۳، ۲، ۴ و ۵ وجود دارند که انتقال همزمان آنها باید بررسی شود. در سطح ۱



شکل ۵- چهار گره از یک شبکه



شکل ۶- شبکه مثال برای الگوریتم تفاضلی

ارزش فعلی کل شبکه می‌شود آن انتقال صورت بگیرد. اما به دلیل اینکه بررسی همه فعالیت‌های بعدی هر گره خصوصاً در شبکه‌های با ضریب پیچیدگی بالا، موجب افزایش زمان محاسبات می‌شود، از آن صرف نظر شد.

۳-۴- حل یک مثال

در این قسمت با ذکر یک مثال به طور دقیقتر با نحوه کارکرد الگوریتم بیان خواهد شد. در شکل (۶) شبکه‌ای با ۱۳ گره و ۱۷ کمان نشان داده شده است. جدول (۱) اطلاعات مربوط به جریان نقدی و زودترین و دیرترین زمان هر گره را نشان می‌دهد. برای حل این مثال با الگوریتم تفاضلی، در گام اول همه گره‌های شبکه را در زودترین زمانشان زمان بندی می‌شود، شکل (۷).

پس از زمان بندی همه گره‌ها در زودترین زمان، خالص ارزش فعلی کل پروژه محاسبه و به عنوان بهترین جواب در این مرحله نگهداری می‌شود.

$$BEST = -11/16$$

ذخیره می‌شود و اگر بهبود پیدا نکند، بدون ذخیره کردن نتایج با برگشت به سطح ۲، حالت غیرفعال انتقال گره ۴ بررسی می‌شود (جعبه ۵). این روال در جعبه‌های شکل (۴) به ترتیب شماره آنها ادامه می‌یابد تا جایی که همه حالات ممکن بین آنها بررسی شود.

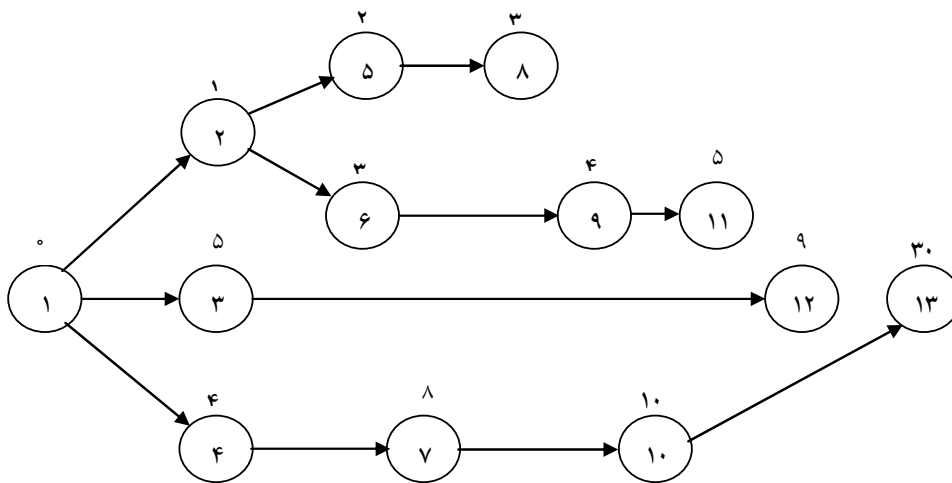
همان گونه که بیان شد. الگوریتم تفاضلی یک الگوریتم ابتکاری است. ذیلاً به یکی از دلایل ابتکاری بودن آن اشاره می‌شود.

شکل (۵) قسمتی از یک شبکه که شامل چهار گره می‌شود، نشان می‌دهد. اگر جریان نقدی گره ۲ به گونه‌ای باشد که قابلیت انتقال دادن گره ۳ به جلو داشته باشد اما نتواند گره ۴ را به جلو انتقال دهد، در این صورت گره ۲ انتقال پیدا نمی‌کند، در صورتی که اگر انتقال گره ۲ و ۳ به اندازه ۴ واحد شناوری فعالیت ۳-۴ صورت گیرد نیازی به انتقال گره ۴ نیز وجود ندارد.

برای رفع این مشکل در الگوریتم، اصلاحاتی به عمل آورده شد، به گونه‌ای که برای هر گره، شناوری‌هایی را که در فعالیت‌های مابعد آن وجود دارد محاسبه شود و اگر باعث بهبود خالص

جدول ۱- اطلاعات شبکه حل شده

| شماره گره | جریان نقدی | دیرترین زمان | زودترین زمان |
|-----------|--------------|--------------|--------------|
| ۱ | ۰ | ۱۰ | ۰ |
| ۲ | $-۳-./۵T$ | ۳۴ | ۱ |
| ۳ | ۵ | ۱۸ | ۸ |
| ۴ | $-۵- T$ | ۱۴ | ۴ |
| ۵ | $-۵- T$ | ۳۷ | ۲ |
| ۶ | $-./۵$ | ۳۶ | ۳ |
| ۷ | $- ./۵$ | ۱۸ | ۸ |
| ۸ | ۱ | ۳۶ | ۳ |
| ۹ | ۱ | ۳۷ | ۴ |
| ۱۰ | ۱ | ۲۰ | ۱۰ |
| ۱۱ | $- ۱ -./۲۵T$ | ۳۸ | ۵ |
| ۱۲ | ۴ | ۳۹ | ۹ |
| ۱۳ | ۰ | ۴۰ | ۳۰ |



شکل ۷- شبکه زمان بندی همه گره ها در زودترین زمان ممکن

$$V = T[12] - D[11-12] - T[11] = 9 - 1 - 5 = 3$$

سپس محاسبه می شود که آیا انتقال زمان گره ۱۱ به اندازه

۳ واحد مفید است یا خیر؟

$$NPV(t=5) = -1.30 < NPV(t=8) = -1.29$$

لذا زمان گره ۱۱ را به ۸ تغییر می دهیم. بعد از این باید

بررسی کنیم که آیا انتقال زمان گره ۱۱ به دیرترین زمان نیز

مفید است یا خیر؟ لذاخالص ارزش فعلی زمان فعلی گره و

سپس بررسی گره ها را از گره ۱۳ شروع می کنیم. چون

جریان نقدی گره ۱۳ صفر است نیازی به تغییر زمان آن وجود

ندارد. گره بعدی گره ۱۲ است که دارای جریان نقدی مثبت

است و به تأخیر انداختن آن خالص ارزش فعلی کل شبکه را

افزایش نمی دهد. برای گره ۱۱ که گره بعدی است حداقل

فاصله این گره با گره های بعدی آن محاسبه می شود.

خالص ارزش فعلی گره در دیرترین زمان محاسبه می‌شود.

$$NPV(t=8) = -1.29 < NPV(Lr=38) = -0.2$$

چون جریان نقدی این گره قابلیت انتقال به جلو را دارد، به طور موقت زمان این گره و گره‌های بعدی آن را افزایش می‌دهیم.

$$T[11] = 38, T[12] = 39, T[13] = 40$$

$$NPV(\text{Project}) = -11.5 < \text{BEST} = -11.16$$

ملاحظه می‌شود که این انتقال خالص ارزش فعلی کل پروژه را افزایش نمی‌دهد. لذا زمان گره‌های ۱۱، ۱۲ و ۱۳ به صورت قبل (به ترتیب ۸، ۹ و ۳۰) حفظ می‌شود. انتقال گره‌های ۱۰، ۹ و ۸، که جریان نقدی ثابت و مثبت دارند نمی‌تواند خالص ارزش فعلی کل پروژه را افزایش دهد لذا بدون بررسی آنها، گره ۷ مورد بررسی قرار می‌گیرد. جریان نقدی این گره ثابت و منفی است لذا به تأخیر انداختن آن خالص ارزش فعلی گره ۷ را افزایش می‌دهد. حداقل فاصله این گره با گره بعدی آن (گره ۱۰)، صفر است لذا انتقال زمان گره ۷ به دیرترین زمانش بررسی می‌شود. زمان گره ۷ و گره‌های بعدی اش به طور موقت انتقال داده می‌شود. گره‌های ۷ و ۱۰ به صورت زیر انتقال پیدا می‌کنند و گره ۱۳ نیز نیاز به انتقال ندارد.

$$T[7] = 18, T[10] = 20$$

$$NPV(\text{Project}) = -11.24 < \text{BEST} = -11.16$$

به دلیل اینکه خالص ارزش فعلی افزایش پیدا نکرده، زمان گره‌های ۷ و ۱۰ بدون تغییر حفظ می‌شوند. گره بعدی گره ۶ است که جریان نقدی منفی و ثابت دارد. حداقل فاصله این گره با گره بعدی (گره ۹)، صفر واحد زمانی است، لذا انتقال به دیرترین زمان ممکن برای گره ۶ مورد بررسی قرار می‌گیرد. گره ۶ به دیرترین زمان ممکن منتقل شده و همه گره‌های بعدی آن به اندازه مورد نیاز به جلو منتقل می‌شوند.

$$T[6] = Lr(6) = 36, T[9] = 37, T[11] = 38, T[12] = 39, T[13] = 40$$

$$NPV(\text{Project}) = -11.82$$

چون خالص ارزش فعلی کل پروژه بهبود پیدا نکرده است، زمان گره‌ها به صورت قبل از انتقال حفظ می‌شوند و گره ۵ مورد بررسی قرار می‌گیرد. برای گره ۵ که جریان نقدی آن به

صورت خطی غیر فزاینده است، ابتدا حداقل فاصله گره ۵ با گره بعدی (گره ۸) محاسبه می‌شود که صفر است. در گام بعد خالص ارزش فعلی گره ۵ در زمان فعلی و در دیرترین زمان، مورد محاسبه قرار می‌گیرد.

$$NPV(T=2) = -5.67 < NPV(Lr=37) = -1$$

بنابراین زمان گره ۵ و گره‌های بعدی آن را موقتاً انتقال داده می‌شوند.

$$T[5]=Lr(5)=35, T[8]=36, T[9]=37, T[11]=38, T[12]=39, T[13]=40$$

$$NPV(\text{Project}) = -8.21$$

چون خالص ارزش فعلی بهبود پیدا کرده است زمان گره‌ها ذخیره شده و بهترین جواب به هنگام می‌شود.

$$\text{BEST} = -8.21$$

گره بعدی که باید بررسی شود گره ۴ است. این گره با گره‌های بعدی فاصله‌ای ندارد، لذا خالص ارزش فعلی آن در زمان فعلی و دیرترین زمان ممکن برای آن محاسبه و مقایسه می‌شود.

$$NPV(T=4) = -5.9 < NPV(Lr=14) = -4.35$$

بنابراین انتقال زمان گره ۴ و گره‌های بعدی آن محاسبه می‌شود.

$$T[4]=14, T[7]=18, T[10]=20$$

$$NPV(\text{Project}) = -6.74$$

خالص ارزش فعلی بهبود یافته است، لذا زمان بندی شبکه و بهترین جواب به هنگام می‌شود.

$$\text{BEST} = -6.74$$

گره ۳ جریان نقدی مثبت دارد، لذا گره ۲ مورد بررسی قرار می‌گیرد. حداقل فاصله این گره با گره‌های بعدی (گره‌های ۵ و ۶) صفر است، لذا انتقال به دیرترین زمان ممکن بررسی می‌شود.

$$NPV(T=1) = -3.15 < NPV(Lr=34) = -.056$$

قابلیت انتقال را دارد لذا:

$$T[2]=Lr(2)=34, T[6]=36$$

بقیه گره‌های بعدی گره ۲ نیازی به تغییر زمان ندارند.

$$NPV(\text{Project}) = -3.79$$

جواب بهبود یافته است لذا نتایج ذخیره می‌شوند. جریان نقدی گره ۱ نیز صفر است و نیازی به انتقال آن نیست. به دلیل اینکه هیچ گره‌ای که قابلیت انتقال داشته باشد و انتقال داده نشده باشد وجود



شکل ۸- نمودار جریان‌ی الگوریتم تفاضلی

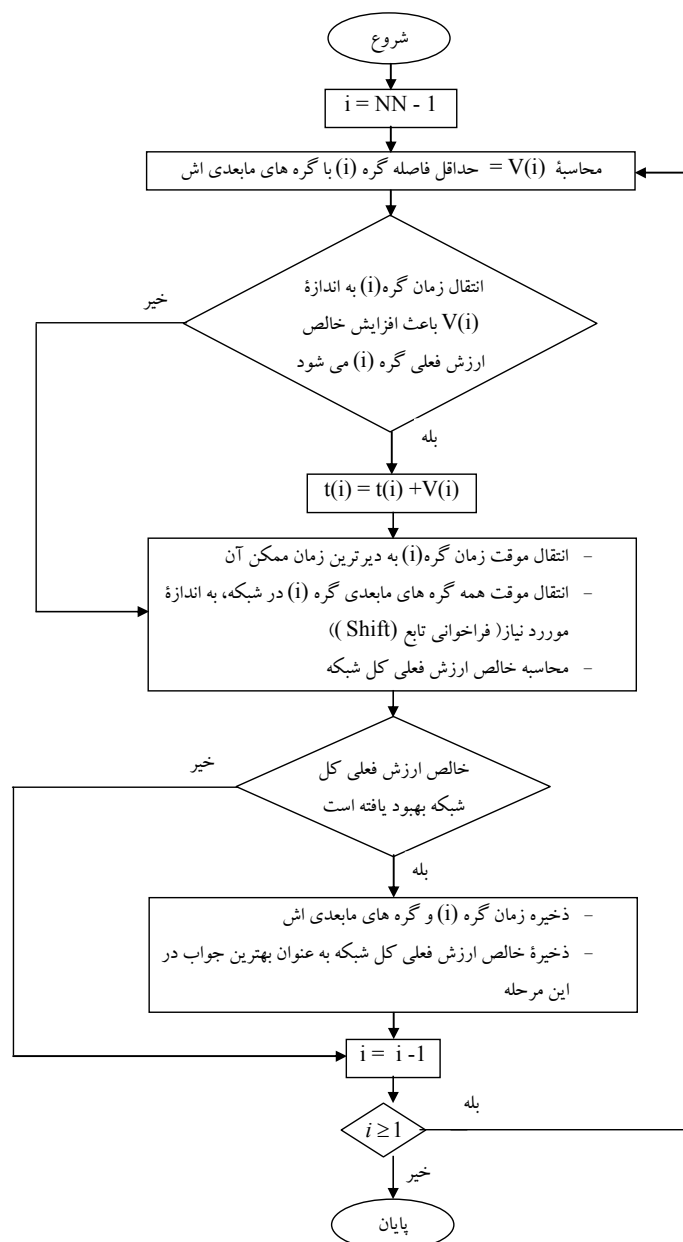
ندارد، الگوریتم پایان می‌یابد. نمودار جریان‌ی این الگوریتم در شکل (۹) به طور جزئی نشان داده شده است.

۴- نتایج محاسباتی

به منظور نشان دادن کارایی الگوریتم تعداد زیادی مسئله حل شده است. آن گونه که در مقالات مختلف آمده است، کارایی هر الگوریتم به چند پارامتر حساسیت دارد و برای درک کارایی هر الگوریتم باید حساسیت آن الگوریتم را نسبت به این پارامترها ارزیابی نمود. پارامترهای مسئله PSP_MAX_NPV شامل تعداد گره‌های شبکه، ضریب پیچیدگی شبکه (نسبت تعداد کمانهای شبکه به تعداد گره‌های آن) و سررسید پروژه است [۹]. برای این منظور از یک سری داده شاخص که قبلاً توسط کولیش و همکارانش تولید شده و از طریق اینترنت قابل

دسترسی بود استفاده شد [۹]. این داده‌ها شامل شبکه‌هایی با ۳۲ و ۶۲ گره و با ضریب پیچیدگی شبکه $1/5$ بود که هر دسته حدود ۱۵۰ مسئله را شامل می‌شد. این مسایل پس از اینکه از طریق برنامه نویسی به فرمت دلخواه درآمدند، برای آزمون الگوریتم تفاضلی و مقایسه آن با الگوریتم برگشتی وانهوک و همکاران مورد استفاده قرار گرفتند.

علاوه بر این مسایل، برای تحلیل و درک دقیق‌تر کارایی الگوریتم تفاضلی، یک سری مسأله تصادفی دیگر نیز تولید شد. تولید مسئله تصادفی شامل تولید شبکه و تولید داده های جریان نقدی است. برای تولید شبکه از یک الگوریتم ابداعی استفاده شده است. نحوه تولید جریانهای براساس روش وانهوک و همکاران است [۸]. در این روش عرض از مبدأ جریان نقدی هر گره (a_i) در دامنه بین $[-50, 50]$ ، و نیز شیب تابع جریان نقدی



شکل ۹- نمودار جریان‌ی گام دوم

حالت (از ۱۰ تا ۲۰۰ گره)، ضریب پیچیدگی شبکه به ۱۸ حالت (از ۱/۵ تا ۶/۶) و مقدار ثابت اضافه شده به مسیر بحرانی شبکه به ۲۰ حالت (از ۵ تا ۱۰۰ واحد) افزایش داده شد. برای هر حالت حدود ۱۰۰ مسئله نمونه و در مجموع ۱۱۶۰۰ مسئله نمونه تولید شد. همچنین برای درک سقف کارایی الگوریتم تفاضلی ۱۰۰ مسئله نمونه با تعداد ۱۰۰۰ گره تولید و حل شد.

هر گره (b_i)، در دامنه [۰ و ۲-]، تولید شده‌اند. آنها همچنین برای تعداد گره مسائل تولید شده ۴ حالت ۳۰، ۶۰، ۹۰ و ۱۲۰ گرهی و برای ضریب پیچیدگی شبکه، ۳ حالت ۱/۵، ۱/۸ و ۲/۱ در نظر گرفتند. سررسید پروژه نیز از اضافه کردن سه مقدار ثابت (۱۵، ۱۰ و ۵) به مسیر بحرانی شبکه به دست می‌آمد. در این مقاله برای تحلیل دقیق‌تر کارایی الگوریتمهای تفاضلی و برگشتی، دامنه تغییرات پارامترهای تعداد گره به ۲۰

جدول ۲- تأثیر تعداد گره در الگوریتم تفاضلی و V_AOA, ($\delta=5$ و $\text{cnc}=1/5$)

| ردیف | تعداد گره | زمان تفاضلی (ثانیه) | زمان V_AOA (ثانیه) | تعداد جوابهای دقیق | درصد اختلاف جواب موارد غیر دقیق |
|------|-----------|---------------------|--------------------|--------------------|---------------------------------|
| ۱ | ۱۰ | ۰/۰۲ | ۰/۱۱ | ۹۹ | ۰/۰۴۶ |
| ۲ | ۲۰ | ۰/۰۴ | ۰/۱۳ | ۱۰۰ | ۰/۰۰۰ |
| ۳ | ۳۰ | ۰/۰۹ | ۰/۱۶ | ۹۹ | ۰/۰۹۶ |
| ۴ | ۴۰ | ۰/۱۲ | ۰/۳۱ | ۹۶ | ۰/۰۹۹ |
| ۵ | ۵۰ | ۰/۱۶ | ۰/۴۴ | ۹۸ | ۰/۰۰۸ |
| ۶ | ۶۰ | ۰/۲۰ | ۰/۵۶ | ۹۸ | ۰/۰۰۹ |
| ۷ | ۷۰ | ۰/۲۶ | ۰/۷۵ | ۹۸ | ۰/۰۴۴ |
| ۸ | ۸۰ | ۰/۳۵ | ۱/۰۲ | ۹۹ | ۰/۰۰۴ |
| ۹ | ۹۰ | ۰/۴۱ | ۱/۵۲ | ۹۸ | ۰/۰۱۱ |
| ۱۰ | ۱۰۰ | ۰/۵۰ | ۲/۰۹ | ۹۹ | ۰/۰۰۴ |
| ۱۱ | ۱۱۰ | ۰/۵۷ | ۲/۹۸ | ۹۶ | ۰/۵۳۰ |
| ۱۲ | ۱۲۰ | ۰/۶۵ | ۳/۴۸ | ۹۷ | ۰/۳۶۹ |
| ۱۳ | ۱۳۰ | ۰/۷۶ | ۴/۸۰ | ۹۸ | ۰/۰۰۶ |
| ۱۴ | ۱۴۰ | ۰/۸۹ | ۶/۱۹ | ۹۵ | ۰/۰۱۱ |
| ۱۵ | ۱۵۰ | ۰/۹۸ | ۷/۷۶ | ۱۰۰ | ۰/۰۰۰ |
| ۱۶ | ۱۶۰ | ۱/۱۰ | ۷/۶۲ | ۹۵ | ۰/۰۳۰ |
| ۱۷ | ۱۷۰ | ۱/۲۱ | ۱۰/۹۶ | ۹۸ | ۰/۰۸۲ |
| ۱۸ | ۱۸۰ | ۱/۳۵ | ۱۳/۲۹ | ۹۹ | ۰/۰۰۳ |
| ۱۹ | ۱۹۰ | ۱/۵۲ | ۱۶/۶۰ | ۹۶ | ۰/۰۱۲ |
| ۲۰ | ۲۰۰ | ۱/۶۲ | *؟ | ؟ | ؟ |
| ۲۱ | ۵۰۰ | ۱۰/۹۵ | ؟ | ؟ | ؟ |
| ۲۲ | ۱۰۰۰ | ۴۱/۶۴ | ؟ | ؟ | ؟ |

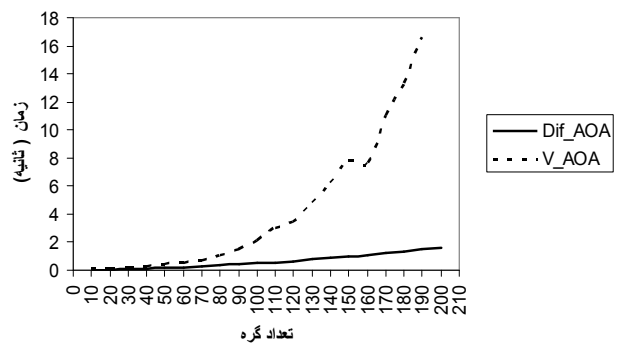
* علامت ؟ نشان دهنده حل نشدن مسائل توسط روش V_AOA به علت نداشتن حافظه است.

تعداد گره بر زمان محاسبات، تأثیر سررسید پروژه بر زمان محاسبات و تأثیر ضریب پیچیدگی شبکه بر زمان محاسبات مورد بررسی قرار می‌گیرد.

۴-۱- تأثیر تعداد گره بر زمان محاسبات

برای بررسی تأثیر تعداد گره بر زمان محاسبات، ۲۲ حالت در نظر گرفته شده است. ضریب پیچیدگی شبکه برابر ۱/۵ و سررسید پروژه به اندازه طول مسیر بحرانی به علاوه ۵، در نظر گرفته شده‌اند. نتیجه اجرای هر دو الگوریتم در جدول (۲) آمده است.

همان گونه که در شکل (۱۰) نشان داده شده است، روش V_AOA نسبت به تعداد گره‌های شبکه حساسیت بالایی دارد، به گونه‌ای که با افزایش تعداد گره‌های شبکه، زمان محاسبات به طور نمایی رشد می‌کند. همچنین این الگوریتم به دلیل اینکه برای اجرا به حافظه زیادی نیاز دارد، برای تعداد گره بیشتر از ۱۹۰ جواب نمی‌دهد. اما روند رشد زمان محاسبات نسبت به



شکل ۱۰- نمودار تأثیر تعداد گره بر زمان محاسبات

برای مقایسه دو الگوریتم تفاضلی (Dif_AOA) و برگشتی وانهوک و همکاران (V_AOA)، این دو الگوریتم، با نرم افزار ویژوال سی ++، نسخه ۷ (Visual Studio.Net)، تحت ویندوز ۲۰۰۰، روی یک رایانه پنتیوم چهار با پردازنده ۱/۶ گیگاهرتز با ۲۵۶ مگابایت RAM، برنامه نویسی و با داده های تولید شده اجرا شدند. در ادامه این قسمت نتایج محاسباتی در سه حالت تأثیر

جدول ۳- تأثیر تعداد گره در الگوریتم تفاضلی و V_AOA ($\delta=5$ و $cnc=30$)

| ردیف | δ | زمان تفاضلی (ثانیه) | زمان V_AOA (ثانیه) | تعداد جوابهای دقیق | درصد اختلاف جواب موارد غیر دقیق |
|------|----------|---------------------|-----------------------|--------------------|---------------------------------|
| ۱ | ۵ | ۰/۰۹ | ۰/۵۵ | ۹۹ | ۰/۰۹۶۰۰۰ |
| ۲ | ۱۰ | ۰/۰۷ | ۰/۴۳ | ۹۷ | ۰/۰۳۵۰۰۰ |
| ۳ | ۱۵ | ۰/۰۷ | ۰/۴۶ | ۹۹ | ۰/۰۱۷۹۰۰ |
| ۴ | ۲۰ | ۰/۰۷ | ۰/۴۵ | ۹۸ | ۰/۰۱۰۰۰۰ |
| ۵ | ۲۵ | ۰/۰۷ | ۰/۴۶ | ۹۸ | ۰/۰۰۸۰۰۰ |
| ۶ | ۳۰ | ۰/۰۸ | ۰/۴۵ | ۹۹ | ۰/۰۰۵۰۰۰ |
| ۷ | ۳۵ | ۰/۰۷ | ۰/۴۷ | ۹۹ | ۰/۰۰۳۰۰۰ |
| ۸ | ۴۰ | ۰/۰۷ | ۰/۴۸ | ۹۹ | ۰/۰۰۲۰۰۰ |
| ۹ | ۴۵ | ۰/۰۷ | ۰/۴۶ | ۹۸ | ۰/۰۰۱۰۰۰ |
| ۱۰ | ۵۰ | ۰/۰۷ | ۰/۴۶ | ۹۹ | ۰/۰۰۰۹۰۰ |
| ۱۱ | ۵۵ | ۰/۰۸ | ۰/۴۷ | ۹۹ | ۰/۰۰۰۵۰۰ |
| ۱۲ | ۶۰ | ۰/۰۷ | ۰/۴۶ | ۹۹ | ۰/۰۰۰۳۰۰ |
| ۱۳ | ۶۵ | ۰/۰۷ | ۰/۴۶ | ۹۹ | ۰/۰۰۰۲۰۰ |
| ۱۴ | ۷۰ | ۰/۰۸ | ۰/۴۶ | ۹۹ | ۰/۰۰۰۱۰۰ |
| ۱۵ | ۷۵ | ۰/۰۷ | ۰/۴۷ | ۹۹ | ۰/۰۰۰۰۸۰ |
| ۱۶ | ۸۰ | ۰/۰۷ | ۰/۴۸ | ۹۹ | ۰/۰۰۰۰۵۰ |
| ۱۷ | ۸۵ | ۰/۰۷ | ۰/۴۷ | ۹۹ | ۰/۰۰۰۰۳۰ |
| ۱۸ | ۹۰ | ۰/۰۸ | ۰/۴۸ | ۹۹ | ۰/۰۰۰۰۲۰ |
| ۱۹ | ۹۵ | ۰/۰۷ | ۰/۴۷ | ۹۹ | ۰/۰۰۰۰۱۰ |
| ۲۰ | ۱۰۰ | ۰/۰۸ | ۰/۴۷ | ۹۹ | ۰/۰۰۰۰۰۵ |

الگوریتم تفاضلی کارایی خوبی برای شبکه های با تعداد گره بالا دارد.

۴-۲- تأثیر سررسید پروژه بر زمان محاسبات

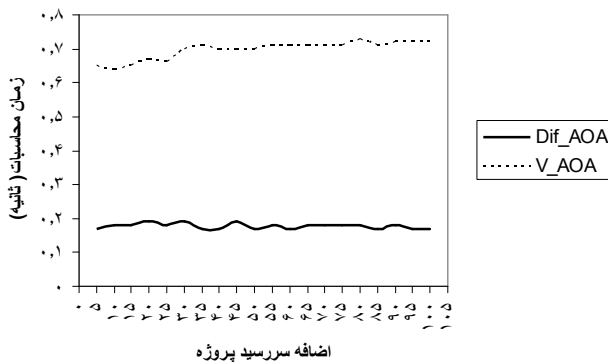
برای بررسی تأثیر سررسید پروژه بر زمان محاسبات، دو حالت در این قسمت مورد بررسی قرار گرفته است. در حالت اول شبکه هایی با تعداد گره ۳۰ و ضریب پیچیدگی شبکه ۱/۵ در نظر گرفته می شود و مقدار اضافه سررسید پروژه نیز بین ۵ تا ۱۰۰ با فواصل ۵ تایی تغییر می کند. نتایج به دست آمده در جدول (۳) و نیز نمودار زمان محاسبات در شکل (۱۱) نشان می دهد که هر دو الگوریتم حساسیت زیادی به تغییر زمان سررسید پروژه نشان نمی دهند. در حالت دوم تعداد گره و ضریب پیچیدگی شبکه به طور هم زمان افزایش می یابند و تعداد گره در ۵۰ و ضریب پیچیدگی شبکه در ۳ در نظر گرفته می شود. جدول (۴) نتایج به دست آمده برای این حالت را نشان می دهد. روند نمودار شکل (۱۲) نشان می دهد که هنگامی که تعداد گره و ضریب پیچیدگی به طور هم زمان

افزایش تعداد گره در الگوریتم تفاضلی بسیار کمتر از الگوریتم وانهوک AOA است. درصد جوابهای دقیق مسایل نمونه حل شده به وسیله الگوریتم تفاضلی، همان گونه که در جدول (۲) مشاهده می شود، بین ۹۵ تا ۱۰۰ درصد متغیر است. روند درصد جوابهای دقیق نشان می دهد که تعداد جوابهای دقیق به دست آمده از الگوریتم تفاضلی حساسیت خاصی نسبت به افزایش تعداد گره ها ندارد، لذا به نظر می آید که در شبکه های با تعداد گره بالاتر نیز بتوان به جوابهای دقیقی در همین حد دست یافت. ستون "درصد اختلاف جواب موارد غیر دقیق" از جدول (۲) نیز درصد اختلاف جواب مسایلی که جواب غیر دقیق داده اند، با جوابهای دقیق الگوریتم V_AOA را نشان می دهد. در این ستون، روند مشخصی از تغییر درصد اختلاف جوابها مشاهده نمی شود.

ردیف ۲۱ و ۲۲ از جدول (۲) تعداد گره ۵۰۰ و ۱۰۰۰ را نشان می دهند. این دو حالت برای ارزیابی سقف کارایی الگوریتم تفاضلی تولید شده اند که زمان محاسبات آنها، به ترتیب حدود ۱۰ و ۴۱ ثانیه است. لذا می توان گفت که

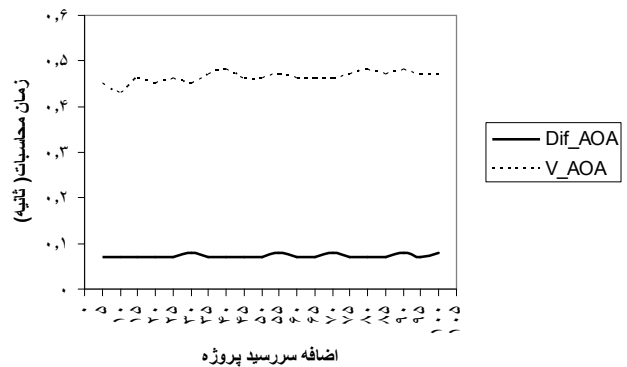
جدول ۴- تأثیر سررسید پروژه در الگوریتم تفاضلی و V_AOA, (cnc = ۳)

| ردیف | δ | زمان تفاضلی (ثانیه) | زمان V_AOA (ثانیه) | تعداد جوابهای دقیق | درصد اختلاف جواب موارد غیر دقیق |
|------|----------|---------------------|--------------------|--------------------|---------------------------------|
| ۱ | ۵ | ۰/۱۷ | ۰/۶۵ | ۹۷ | ۰/۰۰۶۰ |
| ۲ | ۱۰ | ۰/۱۸ | ۰/۶۴ | ۹۹ | ۰/۰۱۵۰ |
| ۳ | ۱۵ | ۰/۱۸ | ۰/۶۵ | ۱۰۰ | ۰/۰۰۰۰ |
| ۴ | ۲۰ | ۰/۱۹ | ۰/۶۷ | ۹۹ | ۰/۰۰۱۰ |
| ۵ | ۲۵ | ۰/۱۸ | ۰/۶۶ | ۹۹ | ۰/۰۱۶۰ |
| ۶ | ۳۰ | ۰/۱۹ | ۰/۷۰ | ۱۰۰ | ۰/۰۰۰۰ |
| ۷ | ۳۵ | ۰/۱۷ | ۰/۷۱ | ۱۰۰ | ۰/۰۰۰۰ |
| ۸ | ۴۰ | ۰/۱۷ | ۰/۷۰ | ۹۹ | ۰/۰۰۰۴ |
| ۹ | ۴۵ | ۰/۱۹ | ۰/۷۰ | ۹۹ | ۰/۰۰۰۲ |
| ۱۰ | ۵۰ | ۰/۱۷ | ۰/۷۰ | ۱۰۰ | ۰/۰۰۰۰ |
| ۱۱ | ۵۵ | ۰/۱۸ | ۰/۷۱ | ۱۰۰ | ۰/۰۰۰۰ |
| ۱۲ | ۶۰ | ۰/۱۷ | ۰/۷۱ | ۱۰۰ | ۰/۰۰۰۰ |
| ۱۳ | ۶۵ | ۰/۱۸ | ۰/۷۱ | ۱۰۰ | ۰/۰۰۰۰ |
| ۱۴ | ۷۰ | ۰/۱۸ | ۰/۷۱ | ۱۰۰ | ۰/۰۰۰۰ |
| ۱۵ | ۷۵ | ۰/۱۸ | ۰/۷۱ | ۱۰۰ | ۰/۰۰۰۰ |
| ۱۶ | ۸۰ | ۰/۱۸ | ۰/۷۳ | ۱۰۰ | ۰/۰۰۰۰ |
| ۱۷ | ۸۵ | ۰/۱۷ | ۰/۷۱ | ۱۰۰ | ۰/۰۰۰۰ |
| ۱۸ | ۹۰ | ۰/۱۸ | ۰/۷۲ | ۱۰۰ | ۰/۰۰۰۰ |
| ۱۹ | ۹۵ | ۰/۱۷ | ۰/۷۲ | ۱۰۰ | ۰/۰۰۰۰ |
| ۲۰ | ۱۰۰ | ۰/۱۷ | ۰/۷۲ | ۱۰۰ | ۰/۰۰۰۰ |



شکل ۱۲ نمودار تأثیر سررسید پروژه بر زمان محاسبات

(NN=50, cnc=3)



شکل ۱۱- نمودار تأثیر سررسید پروژه بر زمان محاسبات

(NN=30 , cnc=1.5)

کرده است. نکته دیگری که در هر دو حالت مشاهده می شود این است که با افزایش مقدار اضافه سررسید پروژه، درصد اختلاف مواردی که جواب غیر دقیق به وسیله الگوریتم تفاضلی به دست آمده است، به سمت صفر میل پیدا می کند. لذا می توان نتیجه گرفت که با افزایش زمان سررسید پروژه و بالطبع افزایش شنواری فعالیتهای شبکه، کارایی الگوریتم تفاضلی بهبود پیدا می کند.

افزایش پیدا کند، زمان محاسبات الگوریتم V_AOA کمی روند صعودی پیدا می کند. اما الگوریتم تفاضلی با وجود افزایش تعداد گره و ضریب پیچیدگی شبکه، همچنان به افزایش زمان سررسید پروژه حساسیت ندارد. از طرف دیگر در حالت اول تعداد جوابهای دقیق الگوریتم تفاضلی به طور میانگین حدود ۹۸/۷۵ درصد است که با افزایش تعداد گره و ضریب پیچیدگی شبکه در حالت دوم، به ۹۹/۶ افزایش پیدا

جدول ۵- تأثیر ضریب پیچیدگی شبکه در الگوریتم تفاضلی و V_AOA ($\delta = 5$ و $NN=30$)

| ردیف | CNC | زمان تفاضلی (ثانیه) | زمان V_AOA (ثانیه) | تعداد جوابهای دقیق | درصد اختلاف جواب موارد غیر دقیق |
|------|-----|---------------------|-----------------------|--------------------|---------------------------------|
| ۱ | ۱/۵ | ۰/۰۷ | ۰/۴۹ | ۹۹ | ۰/۰۹۶ |
| ۲ | ۱/۸ | ۰/۰۸ | ۰/۵۱ | ۹۹ | ۰/۰۰۷ |
| ۳ | ۲/۱ | ۰/۰۸ | ۰/۵۳ | ۹۹ | ۰/۰۰۱ |
| ۴ | ۲/۴ | ۰/۰۸ | ۰/۵۷ | ۹۷ | ۰/۲۳۴ |
| ۵ | ۲/۷ | ۰/۰۹ | ۰/۵۲ | ۱۰۰ | ۰/۰۰۰ |
| ۶ | ۳/۰ | ۰/۰۹ | ۰/۵۶ | ۹۹ | ۰/۰۰۷ |
| ۷ | ۳/۳ | ۰/۰۹ | ۰/۵۵ | ۱۰۰ | ۰/۰۰۰ |
| ۸ | ۳/۶ | ۰/۱۰ | ۰/۵۸ | ۱۰۰ | ۰/۰۰۰ |
| ۹ | ۳/۹ | ۰/۱۰ | ۰/۶۰ | ۹۹ | ۰/۰۹۴ |
| ۱۰ | ۴/۲ | ۰/۱۰ | ۰/۶۱ | ۹۹ | ۰/۰۱۵ |
| ۱۱ | ۴/۵ | ۰/۱۰ | ۰/۵۶ | ۹۸ | ۰/۰۰۷ |
| ۱۲ | ۴/۸ | ۰/۱۱ | ۰/۵۵ | ۱۰۰ | ۰/۰۰۰ |
| ۱۳ | ۵/۱ | ۰/۱۱ | ۰/۵۷ | ۱۰۰ | ۰/۰۰۰ |
| ۱۴ | ۵/۴ | ۰/۱۱ | ۰/۵۷ | ۹۹ | ۰/۰۰۰ |
| ۱۵ | ۵/۷ | ۰/۱۱ | ۰/۶۴ | ۱۰۰ | ۰/۰۰۰ |
| ۱۶ | ۶/۰ | ۰/۱۲ | ۰/۵۶ | ۱۰۰ | ۰/۰۰۰ |
| ۱۷ | ۶/۳ | ۰/۱۳ | ۰/۵۷ | ۱۰۰ | ۰/۰۰۰ |
| ۱۸ | ۶/۶ | ۰/۱۳ | ۰/۵۵ | ۱۰۰ | ۰/۰۰۰ |

جدول ۶- تأثیر ضریب پیچیدگی شبکه در الگوریتم تفاضلی و V_AOA ($\delta = 5$ و $NN=50$)

| ردیف | CNC | زمان تفاضلی (ثانیه) | زمان V_AOA (ثانیه) | تعداد جوابهای دقیق | درصد اختلاف جواب موارد غیر دقیق |
|------|-----|---------------------|-----------------------|--------------------|---------------------------------|
| ۱ | ۱/۵ | ۰/۱۴ | ۱/۰۴ | ۹۸ | ۰/۰۰۸ |
| ۲ | ۱/۸ | ۰/۱۶ | ۱/۲۸ | ۹۷ | ۰/۱۳۹ |
| ۳ | ۲/۱ | ۰/۱۷ | ۱/۵۵ | ۹۷ | ۰/۰۸۵ |
| ۴ | ۲/۴ | ۰/۱۷ | ۱/۴۷ | ۱۰۰ | ۰/۰۰۰ |
| ۵ | ۲/۷ | ۰/۱۷ | ۱/۷۸ | ۹۸ | ۰/۰۴۸ |
| ۶ | ۳/۰ | ۰/۱۷ | ۱/۵۸ | ۹۷ | ۰/۰۰۶ |
| ۷ | ۳/۳ | ۰/۱۹ | ۱/۸۳ | ۹۹ | ۰/۰۱۱ |
| ۸ | ۳/۶ | ۰/۱۸ | ۱/۵۸ | ۱۰۰ | ۰/۰۰۰ |
| ۹ | ۳/۹ | ۰/۲۰ | ۱/۹۶ | ۹۸ | ۰/۰۰۴ |
| ۱۰ | ۴/۲ | ۰/۲۲ | ۱/۵۷ | ۹۹ | ۰/۰۰۸ |
| ۱۱ | ۴/۵ | ۰/۲۰ | ۱/۵۴ | ۹۹ | ۰/۱۱۲ |
| ۱۲ | ۴/۸ | ۰/۲۱ | ۱/۶۹ | ۹۸ | ۰/۱۸۴ |
| ۱۳ | ۵/۱ | ۰/۲۱ | ۱/۶۵ | ۹۸ | ۰/۰۲۶ |
| ۱۴ | ۵/۴ | ۰/۲۲ | ۱/۸۷ | ۹۹ | ۰/۰۱۲ |
| ۱۵ | ۵/۷ | ۰/۲۳ | ۱/۷۲ | ۹۷ | ۰/۰۳۹ |
| ۱۶ | ۶/۰ | ۰/۲۴ | ۱/۷۶ | ۹۷ | ۰/۰۰۷ |
| ۱۷ | ۶/۳ | ۰/۲۴ | ۱/۶۳ | ۹۹ | ۰/۰۰۰ |
| ۱۸ | ۶/۶ | ۰/۲۳ | ۱/۴۶ | ۹۹ | ۰/۰۰۸ |

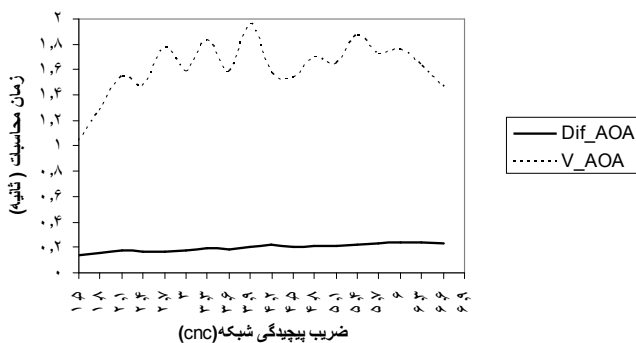
۳-۴- تأثیر ضریب پیچیدگی شبکه بر زمان محاسبات

برای بررسی تأثیر ضریب پیچیدگی شبکه بر زمان محاسبات، تعداد گره‌های شبکه در ۳ حالت ۳۰، ۵۰ و ۷۰ در نظر گرفته شده و سررسید پروژه نیز در هر سه حالت به اندازه مسیر

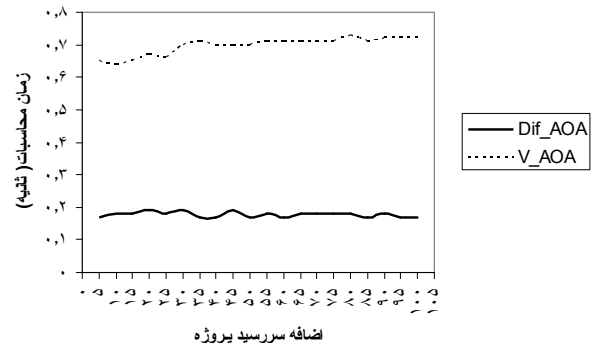
بحرانی به علاوه ۵ در نظر گرفته شده است. جداول (۵) و (۶) و (۷) و نیز نمودارهای (۱۳)، (۱۴) و (۱۵) نتایج به دست آمده را نشان می‌دهد. شکل (۱۳) تأثیر سررسید پروژه بر زمان محاسبات با تعداد گره ۳۰ را نشان می‌دهد. همان گونه که از

جدول ۷- تأثیر ضریب پیچیدگی شبکه در الگوریتم تفاضلی و V_AOA ($\delta=5$ و $\delta=70$)

| ردیف | CNC | زمان تفاضلی (ثانیه) | زمان V_AOA (ثانیه) | تعداد جوابهای دقیق | درصد اختلاف جواب موارد غیر دقیق |
|------|-----|---------------------|-----------------------|--------------------|---------------------------------|
| ۱ | ۱/۵ | ۰/۲۶ | ۱/۹۸ | ۹۸ | ۰/۰۴۴ |
| ۲ | ۱/۸ | ۰/۲۸ | ۲/۸۴ | ۹۹ | ۰/۳۶۲ |
| ۳ | ۲/۱ | ۰/۲۸ | ۳/۰۰ | ۹۸ | ۰/۰۰۶ |
| ۴ | ۲/۴ | ۰/۲۸ | ۵/۸۷ | ۹۹ | ۰/۰۳۳ |
| ۵ | ۲/۷ | ۰/۳۱ | ۳/۸۲ | ۹۸ | ۰/۰۲۱ |
| ۶ | ۳/۰ | ۰/۳۱ | ۷/۷۱ | ۱۰۰ | ۰/۰۰۰ |
| ۷ | ۳/۳ | ۰/۳۵ | ۸/۱۵ | ۱۰۰ | ۰/۰۰۰ |
| ۸ | ۳/۶ | ۰/۳۴ | ۷/۹۵ | ۹۷ | ۰/۴۸۳ |
| ۹ | ۳/۹ | ۰/۳۷ | ۷/۱۰ | ۹۹ | ۰/۰۰۶ |
| ۱۰ | ۴/۲ | ۰/۳۳ | ۷/۸۸ | ۹۷ | ۰/۰۲۰ |
| ۱۱ | ۴/۵ | ۰/۳۳ | ۷/۰۱ | ۹۸ | ۰/۰۳۸ |
| ۱۲ | ۴/۸ | ۰/۳۵ | ۱۰/۳۷ | ۹۸ | ۰/۰۰۹ |
| ۱۳ | ۵/۱ | ۰/۳۶ | ۵/۵۰ | ۹۸ | ۰/۰۲۲ |
| ۱۴ | ۵/۴ | ۰/۳۹ | ۶/۹۶ | ۹۴ | ۰/۰۵۱ |
| ۱۵ | ۵/۷ | ۰/۴۱ | ۴/۰۵ | ۱۰۰ | ۰/۰۰۰ |
| ۱۶ | ۶/۰ | ۰/۴۰ | ۵/۴۱ | ۹۵ | ۰/۰۲۶ |
| ۱۷ | ۶/۳ | ۰/۳۹ | ۹/۸۱ | ۹۸ | ۰/۰۰۸ |
| ۱۸ | ۶/۶ | ۰/۴۷ | ۸/۴۴ | ۹۹ | ۰/۰۰۸ |



شکل ۱۴- نمودار تأثیر ضریب پیچیدگی شبکه بر زمان محاسبات (تعداد گره ۵۰)



شکل ۱۳- نمودار تأثیر ضریب پیچیدگی شبکه بر زمان محاسبات (تعداد گره ۳۰)

الگوریتم V_AOA در شکل‌های (۱۳) و (۱۴) و (۱۵) مشاهده می‌شود، ناشی از این نکته است که در الگوریتم V_AOA در هر کدام از مجموعه گره‌های تحت بررسی، در صورتی که انتقال به اندازه حداقل فاصله باعث بهبود جواب نشود ولی انتقال به اندازه حداکثر فاصله باعث بهبود جواب شود، آن مجموعه به درخت تأخیر اضافه می‌شود و به وسیله تابع شمارش، انتقالش ارزیابی می‌شود. حال هر شبکه‌ای که تعداد درختهای تأخیرش بیشتر شود، چون نیاز به بررسی بیشتری به وسیله تابع شمارش دارد، زمان محاسباتی اش بیشتر می‌شود. از

نمودار استنباط می‌شود همزمان با افزایش ضریب پیچیدگی شبکه زمان محاسبات الگوریتم تفاضلی به طور خفیف افزایش پیدا می‌کند. خط روند زمان محاسباتی الگوریتم V_AOA نیز سیر صعودی دارد، اما به طور دقیق نمی‌توان گفت که روند صعود کدام الگوریتم بیشتر است. اما هنگامی که تعداد گره را به ۵۰ و بعد به ۷۰ افزایش می‌یابد، همان طور که در شکل‌های (۱۴) و (۱۵) مشاهده می‌شود، روند صعود زمان محاسباتی الگوریتم V_AOA به مراتب بیشتر از روند صعود زمان محاسباتی الگوریتم تفاضلی است. نوساناتی که در نمودار

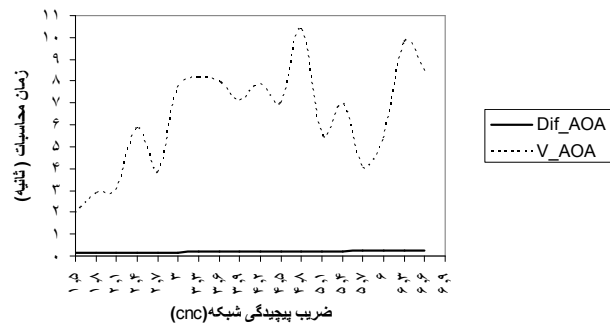
گرفت و الگوریتمی برای آن ارائه شد. نتایج محاسباتی به دست آمده نشان می‌دهد که الگوریتم تفاضلی برای مسایلی با ۱۰۰۰ گره در زمان بسیار کمی جواب می‌دهد، در حالی که الگوریتم V_AOA به دلیل اینکه به حافظه بالایی نیاز دارد برای مسایل بیشتر از ۱۹۰ گره، امکان گرفتن جواب از آن میسر نشد. همچنین نتایج محاسباتی نشان داد که هنگامی که تعداد گره‌ها افزایش می‌یابد، بین ۹۵ تا ۱۰۰ درصد جوابهای الگوریتم تفاضلی دقیق‌اند.

همچنین نشان داده شده که الگوریتم تفاضلی تقریباً هیچ حساسیتی نسبت به زمان سررسید پروژه ندارد و نیز افزایش سررسید پروژه باعث کاهش درصد اختلاف جواب مواردی است که الگوریتم تفاضلی جواب غیر دقیق می‌دهد. الگوریتم V_AOA نیز حساسیت بسیار کمی نسبت به افزایش سررسید پروژه دارد، اما نسبت به الگوریتم تفاضلی حساسیت بیشتری دارد و این حساسیت هر چه تعداد گره‌ها و ضریب پیچیدگی شبکه بالاتر باشد، بیشتر است. همچنین نشان داده شده است که هر دو الگوریتم نسبت به افزایش ضریب پیچیدگی شبکه، حساس‌اند ولی الگوریتم تفاضلی نسبت به الگوریتم V_AOA حساسیت بسیار کمی به افزایش ضریب پیچیدگی شبکه دارد.

در ادامه تحقیق می‌توان علت ابتکاری بودن الگوریتم مورد بررسی قرار گیرد و در صورت امکان بتوان آن را به یک الگوریتم بهینه تبدیل کرد. همچنین مناسب است محدودیت منابع به الگوریتم اضافه شود. همچنین پیدا کردن درجه محاسباتی الگوریتم ارائه شده قضاوت بهتری را برای تشخیص کارایی آن فراهم می‌کند

قدردانی

از داوران محترم که با ارائه نقطه نظرات مناسب، موجب غنای هر چه بیشتر مطالب علمی مقاله شدند، تشکر می‌شود.



شکل ۱۵- نمودار تأثیر ضریب پیچیدگی شبکه بر زمان محاسبات (تعداد گره ۷۰)

طرف دیگر اگر شبکه‌ای وجود داشته باشد که هیچ درخت تأخیری نداشته باشد زمان محاسباتی آن به میزان بسیار زیادی کاهش می‌یابد.

نکته دیگر که از نتایج به دست آمده در این قسمت می‌توان فهمید این است که تعداد جوابهای دقیق به دست آمده به وسیله الگوریتم تفاضلی در حالت ۳۰ گرهی به طور میانگین حدود ۹۹/۳۴ درصد بوده است که با افزایش تعداد گره‌ها به ۵۰ و ۷۰ این مقدار به ترتیب ۹۸/۲۷ و ۹۸/۰۵ شده است. لذا می‌توان گفت که در الگوریتم تفاضلی زمان محاسبات نسبت به افزایش ضریب پیچیدگی شبکه حساسیت کمی دارد و تعداد جوابهای دقیق مقدار کمی کاهش یافته است.

همان طور که ملاحظه می‌شود اختلاف زمان محاسبات بین دو الگوریتم به میزان قابل ملاحظه‌ای در شکل‌های (۱۳)، (۱۴) و (۱۵) بالاست. زمان محاسباتی الگوریتم تفاضلی در هر سه حالت کمتر از ۱ ثانیه قرار گرفته است در حالی که زمان محاسباتی الگوریتم V_AOA با افزایش تعداد گره از ۳۰ به ۷۰ به میزان قابل توجهی افزایش پیدا می‌کند و این افزایش در سطح ضریب پیچیدگی ۴/۸ به حدود ۱۰ برابر نیز می‌رسد.

۵- نتیجه‌گیری و پیشنهادات

در این مقاله مسئله زمان بندی پروژه با معیار خالص ارزش فعلی، که در آن منابع نامحدود فرض شده‌اند، مورد بررسی قرار

- | | |
|---|-----------------------------------|
| 1. complexity network coefficient(CNC) | 6. activity on arc |
| 2. deadline | 7. activity on node |
| 3. project scheduling problem. | 8. milestone |
| 4. project scheduling problem to maximize the net present value | 9. shift |
| 5. simulated annealing | 10. differential _activity on arc |

مراجع

- | | |
|---|---|
| 1. Herroelen, W. S., Dommelen, P.V., and Demeulemeester, E.L., "Theory and Methodology: Project Network Models With Discounted Cash Flows: A Guided Tour through Recent Developments," <i>European Journal of Operational Research</i> , No. 100, 1997. | 6. Etgar, R., Shtub, A., and Leblance, J. L., "Scheduling Project to Maximize Net Present Value –The Case of Time-Dependent, Contingent Cash Flows," <i>European Journal of Operational Research</i> , No. 96, 1996. |
| 2. Shtub, A., and Etgar, R., "A Branch and Bound Algorithm For Scheduling Project To Maximize Net Present Value: The Case Of Time-Dependent, Contingent Cash Flows," <i>International Journal Of Production Research</i> , Vol. 35, No. 12, 1997. | 7. Etgar, R., and Shtub, A., "Scheduling Project Activities to Maximize the Net Present Value – The Case of Linear Time-Dependent Cash flows," <i>International Journal Of Production Research</i> , Vol. 37, No. 2, 1999. |
| 3. Russell, A.H., "Cash Flow in Networks," <i>Management Science</i> . Vol. 16, No. 5, 1970. | 8. Vanhoucke, M., Demulemeester, E., and Herroelen, W. S., "Maximizing the Net Present Value of A Project with Linear Time-Dependent Cash Flows," <i>International Journal of Production Research</i> , Vol. 39, No 14, 2001. |
| 4. Vanhoucke, M., Demulemeester, E., and Herroelen, W. S., "A Validation Of Procedures For Maximizing The Net Present Value Of A Project," <i>Research Report</i> , No. 0030, 2000, Department of Applied Economics, Katolike University Leuven. | 9. Kolisch, R, and Padman, R., "An Integrated Survey of Deterministic Project Scheduling," <i>Omega, the International Journal of Management Science</i> , No. 29, 2001. |
| 5. Elmaghraby, S.E., and Herroelen, W. S., "The Scheduling Of Activities To Maximize The Net Present Value Of Projects," <i>European Journal Of Operational Research</i> , No. 49, 1990. | |