مهندسی صنایع و مدیریت شریف، (تابستان ۱۳۹۵)
دوره‌ی ۱ - ۳۲، شماره‌ی ۱/۲، ص. ۳۵-۴۶

Original Article

ارائه‌ی دو روش فراابتکاری برای مسئله‌ی چندهدفه‌ی موازنه‌ی زمان ـ هزینه ـ کیفیت پروژه در حالت گسسته با محدودیت‌های پیش‌نیازی تعمیم‌یافته

علیرضا عیدی * **(استادیار)**
هیوا فاروقی (استادیار)
فرید عبدی (کارشناس ارشد)
گروه مهندسی صنایع، دانشگاه کردستان

پژوهش‌های انجام‌شده در حوزه‌ی مدیریت پروژه‌ها عمدتاً بر موازنه‌ی زمان ـ هزینه تمرکز داشته‌اند، اما اخیراً عامل کیفیت به‌عنوان یکی از معیارهای اساسی موفقیت پروژه بسیار مورد توجه است. از این‌رو در این تحقیق، مدل جدیدی برای مسئله‌ی موازنه‌ی زمان ـ هزینه ـ کیفیت در حالت گسسته پیشنهاد شده که برخلاف مدل‌های سنتی که در آن‌ها تنها یک نوع رابطه‌ی پیش‌نیازی بین فعالیت‌ها وجود دارد، روابط وابستگی بین فعالیت‌ها از نوع روابط پیش‌نیازی تعمیم‌یافته است و لحاظ کردن این نوع روابط علی‌رغم پیچیده‌تر کردن مسائل، ما را به دنیای واقعیت نزدیک‌تر می‌سازد. به‌دلیل NP-hard بودن این دسته از مسائل و ضرورت استفاده از الگوریتم‌های ابتکاری و فراابتکاری، در این مقاله از دو الگوریتم NSGA-II[1] و FastPGA[2] استفاده شده که کارایی آن‌ها با چندین معیار که بر کیفیت و تنوع جواب‌ها تأکید دارند، مورد مقایسه قرار گرفته است.

واژگان کلیدی: مدیریت پروژه، موازنه‌ی زمان ـ هزینه ـ کیفیت، روابط پیش‌نیازی تعمیم‌یافته، مسائل چندحالته، الگوریتم‌های فراابتکاری NSGA-II و FastPGA.

alireza.eydi@uok.ac.ir
h.farughi@uok.ac.ir
abdi.farid@yahoo.com

۱. مقدمه

زمان کوتاه‌تر، هزینه‌ی کم‌تر و کیفیت بالاتر اهداف اصلی هر پروژه‌اند و همواره بر یکدیگر تأثیرگذارند. یکی از جنبه‌های مهم مدیریت پروژه، درک اطلاعاتی مناسب از تعادل بهینه‌ی این اهداف است.

در اواخر سال ۱۹۵۰ روش مسیر بحرانی (CPM)[3] به‌عنوان ابزاری سودمند در برنامه‌ریزی و زمان‌بندی پروژه‌ها معرفی شد. در محاسبات این روش فرض بر این است که همه‌ی فعالیت‌ها در زمان پیش‌بینی شده و معمول خود قابل انجام است، اما در مواردی لازم می‌شود پروژه حتی زودتر از زمان برنامه‌ریزی شده به اتمام برسد. این تاریخ معمولاً از طرف کارفرما یا مدیریت رده بالا براساس اهداف یا سیاست‌های مختلف تعیین می‌شود. برای دست‌یابی به زمان تکمیل زودتر، باید زمان تعدادی از فعالیت‌ها را کاهش داد. این کاهش زمان توأم با افزایش منابع کاری و صرف هزینه است که آن را انجام ضربتی یا «فشرده‌سازی زمان فعالیت[4]» می‌گویند. از طرف دیگر، انجام فعالیت‌ها در زمان طولانی‌تر، معمولاً سبب کاهش هزینه‌های فعالیت می‌شود اما ممکن است به افزایش زمان پروژه که احتمالاً جریمه‌هایی در بر دارد منجر شود. تصمیم جامع و دقیق در ارتباط با این منافع و جریمه‌ها برای مدیران چالشی سخت است؛ به‌همین دلیل موازنه‌ی زمان ـ هزینه مورد توجه قرارگرفت.

در عمل، یکی از معیارهای اساسی برای موفقیت پروژه کیفیت انجام آن است که ممکن است تحت تأثیر تسریع زمان تکمیل پروژه با هزینه‌های اضافی قرارگیرد. هدف مسائل موازنه‌ی زمان ـ هزینه ـ کیفیت انتخاب مجموعه‌یی از فعالیت‌ها برای تسریع و همچنین انتخاب روش اجرای مناسب برای هر فعالیت است، به‌نحوی که هزینه و زمان پروژه کمینه و کیفیت آن بیشینه شود.

از سوی دیگر، در مدل‌های سنتی CPM فقط یک نوع رابطه بین فعالیت‌های سری وجود دارد. تمامی فعالیت‌هایی که به یک رویداد می‌رسند باید به‌طور کامل اجرا شده‌باشند تا فعالیت‌هایی خروجی از آن رویداد قابل شروع شدن باشند؛ این نوع ارتباط زمانی را ارتباط پایان به آغاز می‌نامند. در اجرای پروژه‌ها علاوه بر این ارتباط ممکن است سه نوع ارتباط زمانی دیگر بین فعالیت‌ها وجود داشته باشد. به هریک از این رابطه‌ها می‌توان یک مقدار عددی نیز اضافه کرد. این مقدار عددی عبارت است از فاصله‌ی زمانی که باید بین لحظات «پایان به آغاز»، «پایان به پایان»، «آغاز به آغاز»، یا «آغاز به پایان» دو فعالیت وجود داشته باشد. این مقدار عددی را «وقفه[5]» می‌نامند.

برای حل مسائل چندهدفه تکنیک‌های متعددی وجود دارد که در دو گروه «روش‌های عددی» و «روش‌های پارتویی» دسته‌بندی می‌شوند. روش‌های عددی

---

* نویسنده مسئول
تاریخ: دریافت ۱۳۹۲/۱۲/۱۲، اصلاحیه ۱۳۹۳/۹/۲۱، پذیرش ۱۳۹۳/۱۲/۱۰.

مسائل چندهدفه را با استفاده از تبدیلات ریاضی به مسائل تک‌هدفه تبدیل می‌کنند. روش‌های پارتو از مفهوم مجموعه‌های غالب برای یافتن جواب‌های پارتویی استفاده می‌کنند. از آنجا که مسئله‌ی موازنه‌ی زمان ـ هزینه ـ کیفیت در حالت گسسته در رده‌ی مسائل NP-hard قرار دارد، با افزودن محدودیت‌های پیش‌نیازی تعمیم‌یافته بر پیچیدگی مسئله افزوده می‌شود. لذا برای به دست آوردن جواب‌های قابل قبول در زمان‌های منطقی باید از الگوریتم‌های ابتکاری یا فراابتکاری برای حل آن استفاده شود. بدین منظور در نوشتار حاضر برای حل مسئله از دو الگوریتم NSGA-II و FastPGA استفاده شده است که نتایج حاصل از آن‌ها با بهره‌گیری از یک سری شاخص‌ها که برکیفیت و گسترش جواب‌ها تأکید دارند مقایسه شده است.

ساختار نوشتار حاضر چنین است: بخش دوم مقاله به مرور ادبیات اختصاص دارد و در بخش سوم، فرمول‌بندی مدل مسئله ارائه شده است. در بخش چهارم روش‌های ممکن برای حل مدل، و نیز معیارهای ارزیابی و مقایسه‌یی این روش‌ها تشریح می‌شود. در بخش پنجم به نتایج محاسباتی خواهیم پرداخت و نهایتاً در بخش ششم جمع‌بندی و نتیجه‌گیریِ حاصله ارائه می‌شود.

۲. پیشینه‌ی تحقیق

با فرض تغییر هزینه‌ی مستقیم یک فعالیت به‌دنبال تغییر زمان اجرای آن، مدل‌های برنامه‌ریزی ریاضی برای کمینه‌کردن هزینه‌های مستقیم توسعه یافته است. این مسائل در ادبیات موضوع به‌عنوان مسائل موازنه‌ی زمان ـ هزینه‌ی پیوسته شناخته می‌شود. در مطالعه‌ی نخستینِ این مسئله،[۱] یک رابطه‌ی خطی بین زمان و هزینه‌ی یک فعالیت در نظرگرفته شد و یک مدل ریاضی و همچنین یک الگوریتم هیوریستیک برای حل آن پیشنهاد شد. سپس روش حلی برای یافتن منحنی موازنه‌ی زمان ـ هزینه ارائه شد[۲] که در آن بهترین زوج‌های مربوط به زمان و هزینه‌ی هر فعالیت برای کمینه‌کردن هزینه‌ی کل پروژه با محدودیت موعد تحویل را می‌توان یافت. شکل‌های دیگری از توابع هزینه‌ی فعالیت نیز مورد مطالعه قرارگرفت. در این مطالعات توابع به‌صورت محدب و مقعر در نظرگرفته شد[۳] و منحنی هزینه ـ زمان با قطعه‌های خطی تقریب زده شد و برای حل آن کوشیدند. ایده‌ی تقریب منحنی هزینه ـ زمان مقعر با قطعه‌های خطی[۴] نیز در دستورالعمل شاخه و کران مورد استفاده قرارگرفت.

در بسیاری از موارد عملی، منابع در واحدهای گسسته در دسترس‌اند. مسائل مربوط به این حوزه در ادبیات موضوع، تحت عنوان مسائل با چندین حالت اجرا برای فعالیت‌ها یا مسائل موازنه‌ی زمان ـ هزینه‌ی گسسته (DTCTP)[۶] شناخته می‌شود. مسئله‌ی DTCTP اولین‌بار توسط هیندلانگ و موث[۵] معرفی شد و برای چندین سال مورد توجه قرارگرفت؛ این مسئله یک مسئله‌ی NP-hard است.[۶] در مطالعات اخیر در زمینه‌ی DTCTP توجه زیادی به الگوریتم‌های حل ــ که به دو دسته الگوریتم‌های دقیق و الگوریتم‌های ابتکاری دسته‌بندی می‌شود ــ شده است. الگوریتم‌های دقیق مبتنی بر برنامه‌ریزی پویا، الگوریتم‌های شمارشی یا الگوریتم‌های شاخه و کران هستند. نمونه‌یی از یک الگوریتم فراابتکاری برای حل مسئله‌ی DTCTP مورد مطالعه‌ی محققین قرار گرفته است.[۷] بعدها محققین دیگری نیز به توسعه‌ی مدل ریاضی DTCTP پرداختند. برای مسائل استاندارد چندحالته یک مدل برنامه‌ریزی صفر و ۱ معرفی شد[۸] که در آن، برخلاف مسائل موازنه‌ی زمان ـ هزینه، از چندین منبع (تجدیدپذیر و تجدیدناپذیر) استفاده می‌شود. ادبیات موجود شامل تعدادی تلاش‌های پژوهشی برای توسعه‌ی مسائل چندحالته‌ی استاندارد با روابط پیش‌نیازی تعمیم‌یافته (GPRs)[۷] است؛ برای این مسئله یک فرمولاسیون برنامه‌ریزی ریاضی نیز ارائه شده است.[۹] محققین محدودیت‌های زمان تعویض[۸] را در مسئله‌ی موازنه‌ی زمان ـ هزینه تعریف کردند[۱۰] که با لحاظ‌کردن شیفت‌های کاری مختلف، زمان‌های شروع خاصی را به مسئله تحمیل می‌کند.

بابو و سورش[۱۱] نخستین محققانی بودند که بیان کردند کیفیت کلی انجام پروژه ممکن است تحت تأثیر تسریع پروژه قرار گیرد. آن‌ها فرض کردند که هزینه و کیفیت هر فعالیت به‌صورت خطی با تغییر در زمان تکمیل پروژه تغییر می‌یابد. آن‌ها هر هدف را با اختصاص سطوح مطلوب روی اهداف دیگر بهینه ساختند. بعدها مدل پیشنهادی در یک پروژه‌ی واقعی ساخت کارخانه‌ی سیمان در تایلند به کار گرفته شد.[۱۲] با بررسی مسئله‌ی موازنه‌ی زمان ـ هزینه ـ کیفیت در حالت گسسته (DTCQTP)[۹]، محققین از یک مثال دنیای واقعی استفاده کردند[۱۳] و تابع جدیدی برای لحاظ‌کردن کیفیت ساخت در مسائل بهینه‌سازی زمان، هزینه و کیفیت در صنایع ساخت‌وساز پیشنهاد کردند. مسئله‌ی DTCQTP بار دیگر مورد بررسی قرار گرفت[۱۴] و مدل‌های برنامه‌ریزی خطی صفر و ۱ وابسته به هم، با فرض این که هر فعالیت در حالت‌های مختلف ممکن است انجام گیرد و همچنین زمان و کیفیت هر فعالیت گسسته و تابع غیرافزایشی از یک منبع تجدیدناپذیراست، توسعه یافت. بعدها الگوریتم‌های فراابتکاری مختلفی برای حل این مسئله توسعه یافت. همچنین متاهیوریستیکی مبتنی بر الگوریتم ژنتیک برای حل این مسئله و یافتن جواب‌های پارتویی پیشنهاد شد.[۱۵] عده‌یی از محققین نیز مسئله‌ی موازنه‌ی زمان ـ هزینه‌ی گسسته با محدودیت کیفیت و منابع را در نظرگرفتند[۱۶] که شامل محدودیت‌های کیفیت، منابع تجدیدپذیر و تجدیدناپذیر می‌شد. عده‌یی دیگر نیز با استفاده از برنامه‌ریزی تصادفی چندهدفه[۱۷] به حل مسئله‌ی موازنه‌ی زمان ـ هزینه ـ کیفیت پرداختند. در این راستا، یک رویکرد بهینه‌سازی فازی چندهدفه ازدحام ذرات را برای حل مسئله‌ی موازنه‌ی زمان ـ هزینه ـ کیفیت پیشنهاد شد[۱۸] که در آن زمان، هزینه و کیفیت به‌وسیله‌ی اعداد فازی توصیف شده است. همچنین مدلی برای DTCQTP توسعه یافت[۱۹] که در آن زمان و هزینه‌ی هر حالت به‌وسیله‌ی اعداد کلاسیک، و کیفیت آن به‌وسیله‌ی متغیرهای زبانی توصیف می‌شود. برای حل آن مدل یک الگوریتم جدید فراابتکاری ژنتیک هیبریدی توسعه داده شد.

۳. فرمول‌بندی مسئله

در اینجا پروژه به‌عنوان یک گراف جهت‌دار $G(V,E)$ تعریف می‌شود که در آن V بیان‌گر مجموعه‌ی گره‌ها (فعالیت‌های پروژه) و E بیان‌گر مجموعه‌ی کمان‌ها (روابط پیش‌نیازی فعالیت‌ها) است. پروژه با شبکه‌ی گرهی (AON) نمایش داده می‌شود. مجموعه‌ی کمان‌های E خود به ۸ دسته $E_{SS}^{\min}$، $E_{SS}^{\max}$، $E_{SF}^{\min}$، $E_{SF}^{\max}$، $E_{FS}^{\min}$، $E_{FS}^{\max}$، $E_{FF}^{\min}$، $E_{FF}^{\max}$ تقسیم می‌شود که نشان‌دهنده‌ی مجموعه‌ی کمان‌های با روابط پیش‌نیازی شروع به شروع، شروع به پایان، پایان به شروع و پایان به پایان از نوع کمینه یا بیشینه هستند.

برای فعالیت $i \in V$ در پروژه، M_i مجموعه‌ی حالت‌های مختلف اجرای فعالیت i است که در آن برای هر حالت اجرا مانند k، یک ترکیب سه‌تایی (t,c,q) معرفی می‌شود که به‌ترتیب بیان‌گر زمان، هزینه و کیفیت یک فعالیت در آن حالت اجراست، به‌گونه‌یی که $t \in Z$، $c \in Z$، $0 \le q \in Z \le 100$ و Z مجموعه‌ی اعداد صحیح است.

با این فرض که اگر r و k دو حالت اجرا برای انجام فعالیت i باشند به‌طوری که $r < k$، آنگاه $t_{ik} > t_{ir}$، $c_{ik} < c_{ir}$ و $q_{ik} \neq q_{ir}$، آنگاه هدف دست‌یابی به

ترکیب بهینه (t_{ik}, c_{ik}, q_{ik}) مربوط به هر فعالیت است به‌نحوی که زمان و هزینه‌ی پروژه کمینه و کیفیت آن بیشینه شود. کیفیت انجام هر فعالیت تابعی از تعدادی شاخص کیفیت است که وزن هرکدام از این شاخص‌ها در حالت‌های مختلف اجرای یک فعالیت متفاوت است. مجموع وزنی این شاخص‌ها، کیفیت کلی حالت اجرای فعالیت را نشان می‌دهد. روابط بین فعالیت‌ها را از نوع روابط پیش‌نیازی تعمیم‌یافته، و زمان و کیفیت هر فعالیت را به‌صورت گسسته و تابعی از یک منبع تجدیدناپذیر در نظر می‌گیریم. نحوه‌ی نمادگذاری برای بیان مدل ریاضی در ادامه تشریح شده است.

۱.۳. مجموعه‌ی اندیس‌ها

V: مجموعه‌ی گره‌ها (فعالیت‌ها)، $V = \{۱, ۲, ..., n\}$؛

E: مجموعه‌ی کمان‌ها؛

$E_{SS}^{\min}$: مجموعه‌ی حاصل از روابط پیش‌نیازی SS از نوع کمینه؛

$E_{SS}^{\max}$: مجموعه‌ی حاصل از روابط پیش‌نیازی SS از نوع بیشینه؛

$E_{SF}^{\min}$: مجموعه‌ی حاصل از روابط پیش‌نیازی SF از نوع کمینه؛

$E_{SF}^{\max}$: مجموعه‌ی حاصل از روابط پیش‌نیازی SF از نوع بیشینه؛

$E_{FS}^{\min}$: مجموعه‌ی حاصل از روابط پیش‌نیازی FS از نوع کمینه؛

$E_{FS}^{\max}$: مجموعه‌ی حاصل از روابط پیش‌نیازی FS از نوع بیشینه؛

$E_{FF}^{\min}$: مجموعه‌ی حاصل از روابط پیش‌نیازی FF از نوع کمینه؛

$E_{FF}^{\max}$: مجموعه‌ی حاصل از روابط پیش‌نیازی FF از نوع بیشینه؛

M_i: مجموعه‌ی حالت‌های اجرا برای فعالیت i، $i \in V$.

۲.۳. پارامترها

t_{ik}: زمان انجام فعالیت i در حالت k، $k = ۱, ..., |M_i|$؛

c_{ik}: هزینه‌ی انجام فعالیت i در حالت k، $k = ۱, ..., |M_i|$؛

q_{ikl}: شاخص کیفیت l در انجام فعالیت i در حالت k، $k = ۱, ..., |M_i|$، $l = ۱, ..., L$؛

Q_i: کران پایین کیفیت فعالیت i؛

w_i: وزن فعالیت i؛

w_{il}: وزن شاخص کیفیت l برای فعالیت i؛

C_I: هزینه‌های غیر مستقیم پروژه؛

$SS_{ij}^{\min}$: رابطه‌ی پیش‌نیازی SS بین فعالیت‌های i و j از نوع کمینه؛

$SS_{ij}^{\max}$: رابطه‌ی پیش‌نیازی SS بین فعالیت‌های i و j از نوع بیشینه؛

$SF_{ij}^{\min}$: رابطه‌ی پیش‌نیازی SF بین فعالیت‌های i و j از نوع کمینه؛

$SF_{ij}^{\max}$: رابطه‌ی پیش‌نیازی SF بین فعالیت‌های i و j از نوع بیشینه؛

$FS_{ij}^{\min}$: رابطه‌ی پیش‌نیازی FS بین فعالیت‌های i و j از نوع کمینه؛

$FS_{ij}^{\max}$: رابطه‌ی پیش‌نیازی FS بین فعالیت‌های i و j از نوع بیشینه؛

$FF_{ij}^{\min}$: رابطه‌ی پیش‌نیازی FF بین فعالیت‌های i و j از نوع کمینه؛

$FF_{ij}^{\max}$: رابطه‌ی پیش‌نیازی FF بین فعالیت‌های i و j از نوع بیشینه.

۳.۳. متغیرهای تصمیم

S_i: زمان شروع فعالیت i، $i \in V$؛

x_{ik}: اگر فعالیت i در حالت k انجام شود $xik = ۱$ و در غیر این صورت $x_{ik} = ۰$.

مدل مسئله‌ی مورد نظر در قالب یک مدل برنامه‌ریزی عدد صحیح چنین بیان می‌شود:

$$\min T = S_n + \sum_{k=۱}^{|M_i|} t_{nk} x_{nk} \quad (۱)$$

$$\min \sum_{i=۱}^{n} \sum_{k=۱}^{|M_i|} c_{ik} x_{ik} + C_I T \quad (۲)$$

$$\max \sum_{i=۱}^{n} w_i \sum_{l=۱}^{L} \sum_{k=۱}^{|M_i|} w_{il} q_{ikl} x_{ik} \quad (۳)$$

s.t.

$$\sum_{k=۱}^{|M_i|} x_{ik} = ۱ \qquad \forall i \in V \quad (۴)$$

$$S_i + SS_{ij}^{\min} \le S_j \qquad \forall (i,j) \in E_{SS}^{\min} \quad (۵)$$

$$S_i + SS_{ij}^{\max} \ge S_j \qquad \forall (i,j) \in E_{SS}^{\max} \quad (۶)$$

$$S_i + SF_{ij}^{\min} \le S_j + \sum_{k=۱}^{|M_j|} t_{jk} x_{jk} \qquad \forall (i,j) \in E_{SF}^{\min} \quad (۷)$$

$$S_i + SF_{ij}^{\max} \ge S_j + \sum_{k=۱}^{|M_j|} t_{jk} x_{jk} \qquad \forall (i,j) \in E_{SF}^{\max} \quad (۸)$$

$$S_i + FS_{ij}^{\min} + \sum_{k=۱}^{|M_i|} t_{ik} x_{ik} \le S_j \qquad \forall (i,j) \in E_{FS}^{\min} \quad (۹)$$

$$S_i + FS_{ij}^{\max} + \sum_{k=۱}^{|M_i|} t_{ik} x_{ik} \ge S_j \qquad \forall (i,j) \in E_{FS}^{\max} \quad (۱۰)$$

$$S_i + FF_{ij}^{\min} + \sum_{k=۱}^{|M_i|} t_{ik} x_{ik} \le S_j + \sum_{k=۱}^{|M_j|} t_{jk} x_{jk} \qquad \forall (i,j) \in E_{FF}^{\min} \quad (۱۱)$$

$$S_i + FF_{ij}^{\max} + \sum_{k=۱}^{|M_i|} t_{ik} x_{ik} \ge S_j + \sum_{k=۱}^{|M_j|} t_{jk} x_{jk} \qquad \forall (i,j) \in E_{FF}^{\max} \quad (۱۲)$$

$$S_i \ge ۰ \qquad \forall i \in V, Integer \quad (۱۳)$$

$$x_{ik} \in \{۰, ۱\} \qquad \forall i \in V, k \in M_i \quad (۱۴)$$

در این مدل سه تابع هدف داریم که آن‌ها را به‌موازات هم بهینه می‌کنیم. تابع هدف اول یا همان رابطه‌ی ۱، زمان کل انجام پروژه و هدف دوم (رابطه‌ی ۲) کل هزینه‌های پروژه شامل هزینه‌های مستقیم و هزینه‌های غیرمستقیم را کمینه‌می‌کند. تابع هدف ۳ کیفیت کلی انجام پروژه را با توجه به شاخص‌های کیفیت، وزن این شاخص‌ها و همچنین میزان اهمیت هر فعالیت بیشینه‌می‌کند.

محدودیت ۴ تضمین می‌کند که یک و فقط یک حالت اجرا به هر فعالیت تخصیص یابد. محدودیت‌های ۵ تا ۱۲ محدودیت‌های پیش‌نیازی تعمیم‌یافته را نشان می‌دهند. محدودیت ۱۳ تضمین می‌کند که متغیر تصمیم S_i یک مقدار صحیح غیر صفر می‌گیرد و محدودیت ۱۴ بیان‌گر صفر و ۱ بودن متغیر x_{ik} است. لازم به ذکر است که متغیر تصمیم S_i وابسته به متغیر تصمیم x_{ik} است و با مشخص شدن حالت‌های اجرای فعالیت‌ها، زمان‌های شروع آن‌ها از طریق روابط پیش‌نیازی به دست می‌آید.

۴. روش‌های ارزیابی

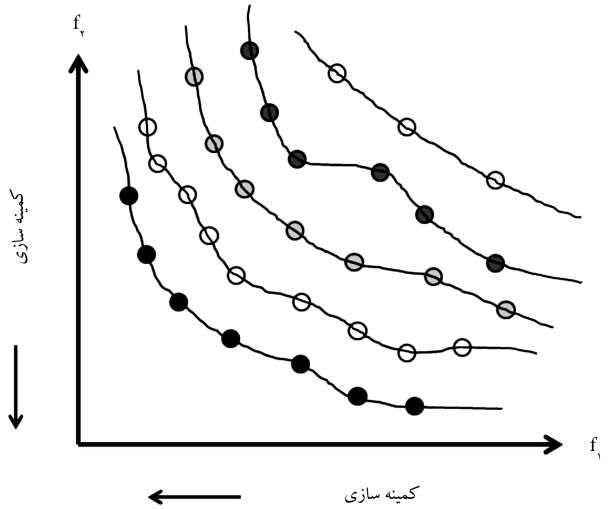
روش‌های حل مسائل چندهدفه به روش‌های کلاسیک و تکاملی تقسیم می‌شوند. نقاط ضعف روش‌های کلاسیک عبارت است از: دست‌یابی به تنها یک جواب بهینه در هر مرحله و نیافتن تمامی جواب‌های بهینه در بهینه‌یابی چندهدفه. برای غلبه بر این موضوع، محققین از روش‌های تکاملی که قابلیت یافتن چندین راه حل بهینه در یک اجرا را دارند، استفاده می‌کنند. الگوریتم‌های ژنتیک خانواده‌یی از الگوریتم‌های ابتکاری برمبنای جست‌وجوی تصادفی و ابزاری قوی برای حل مسائل بهینه‌سازی بزرگ‌مقیاس‌اند. از بین الگوریتم‌های ژنتیک، الگوریتم ژنتیک مرتب‌سازی نامغلوب ۲ به‌دلیل پیچیدگی محاسباتی کم‌تر و استفاده از عملگر فاصله‌ی ازدحام در حل مسائل چندهدفه از کارایی بالایی برخوردار است. همچنین بررسی‌ها نشان داد[۲۰] که روش FastPGA در مسائل بزرگ عملکرد بهتری نسبت به NSGA-II دارد. در ادامه به تشریح هر دو روش پرداخته‌ایم.

۱.۴. الگوریتم NSGA-II

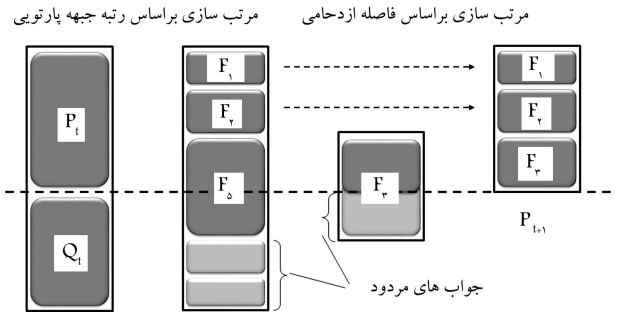
در روش‌های هوشمند و تکاملی برخلاف روش‌های پردازش عددی می‌توان با یک‌بار اجرا مسئله‌ی بهینه‌سازی چندهدفه را حل کرد. از آنجا که در مسائل چندهدفه امکان بهینگی یک راه حل واحد در تمام اهداف و به‌طور همزمان وجود ندارد، الگوریتمی که تعداد راه حل‌هایی را روی پارتو یا نزدیک به پارتو ارائه می‌دهد ارزش عملی بالایی دارد. در واقع مشکلی که در بحث بهینه‌سازی چندهدفه وجود دارد بحث مرتب‌سازی جواب‌هاست که با استفاده از الگوریتم NSGA این مشکل رفع شد. این الگوریتم فضای بهینه‌سازی چندهدفه را که یک فضای مرتب‌پذیر نیست به یک فضای مرتب‌شدنی تبدیل می‌کند. بعدها نسخه‌ی دوم الگوریتم بهینه‌سازی ژنتیک نامغلوب یا NSGA-II ارائه شد[۲۱] که در آن علاوه بر کیفیت جواب‌ها، تنوع و گوناگونی جواب‌های بهینه‌ی پارتویی نیز مورد توجه قرار گرفت.

الگوریتم NSGA-II دو فاز شناخته شده دارد، فاز اول در رابطه با کیفیت جواب‌ها و فاز دوم مربوط به نظم آن‌هاست. در فاز اول رتبه‌بندی جواب‌ها تعیین می‌شود که همان شناسایی جبهه‌های[۱۰] مختلف است؛ برای این امر دو مقدار محاسبه می‌شود: ۱. تعداد دفعاتی که یک جواب مغلوب می‌شود؛ ۲. مجموعه‌ی جواب‌هایی که جواب فعلی بر آن‌ها غلبه می‌کند. برای تعیین این دو مقدار باید تمامی جواب‌ها با یکدیگر مقایسه شود. جواب‌هایی که تعداد دفعات مغلوب شدن‌شان صفر است جواب‌های نامغلوب و تقریبی از جبهه‌ی پارتو هستند؛ این جواب‌ها را جبهه‌ی اول (F_1) می‌نامیم. برای شناسایی جواب‌های جبهه‌ی دوم، ابتدا از تعداد دفعات مغلوب‌شدن همه‌ی جواب‌ها مقدار ۱ را کم می‌کنیم، سپس جواب‌هایی را که مقدار مربوط به تعداد دفعات مغلوب‌شدن‌شان به صفر رسیده به‌عنوان جبهه‌ی دوم در نظر می‌گیریم. این فرایند تا شناسایی تمامی جبهه‌ها تکرار می‌شود (شکل ۱).

در فاز دوم از معیار فاصله‌ی ازدحامی[۱۱] که نشان‌گر فاصله‌ی بین تمامی جواب‌های هم‌سطح (جواب‌های واقع در یک جبهه) است، استفاده می‌کنیم. هرقدر فاصله‌ی ازدحامی نقاط انتخابی بیشتر باشد (در نواحی کم‌جمعیت‌تری قرار گرفته باشند)، این نقاط به تنوع[۱۲] کمک بیشتری می‌کنند. در مقایسه‌ی دو جواب مختلف، با دو حالت مواجه‌ایم: ۱. بین دو راه‌حل با رتبه‌های مختلف، راه‌حل با رتبه‌ی پایین‌تر برتری دارد؛ ۲. اگر دو راه‌حل متعلق به یک جبهه باشند، راه‌حل با مقدار فاصله‌ی



شکل ۱. صفوف مختلف جواب‌های نامغلوب.



شکل ۲. سازوکارکلی عملکرد NSGA-II در تولید نسل بعدی.

ازدحامی بیشتر برگزیده می‌شود.

اگر P_t جمعیت نسل فعلی و Q_t جمعیت فرزندان باشد که با استفاده از عملگرهای تقاطع و جهش ایجاد شده‌اند، برای ایجاد نسل بعدی یا P_{t+1}، ابتدا جواب‌های P_t و Q_t با هم ادغام می‌شوند و سپس روی این جمعیت ادغامی با استفاده از عملگر رتبه‌بندی و پس از آن فاصله‌ی ازدحامی، مرتب‌سازی انجام می‌گیرد. در نهایت به تعداد اندازه جمعیت از جواب‌های مرتب‌شده‌ی اول مستقیماً به نسل بعد منتقل می‌شود و مابقی جواب‌ها حذف می‌شود. درواقع الگوریتم با استفاده از اهمیت بالایی که به رتبه‌ی جواب‌ها و اهمیت نسبی کم‌تری که به فاصله‌ی ازدحامی می‌دهد، بین کیفیت و نظم توازن برقرار می‌کند. در شکل ۲ شمای کلی تولید نسل بعدی در الگوریتم NSGA-II نشان داده شده است.

۲.۴. الگوریتم FastPGA

این الگوریتم برای بهینه‌سازی همزمان چندین هدف که در آن‌ها ارزیابی جواب‌ها به‌لحاظ محاسباتی یا مالی هزینه‌بر هستند، ارائه شد. در این الگوریتم عملگرهای ژنتیکی جدیدی به‌منظور ارتقای عملکرد آن در ارتباط با نحوه‌ی رتبه‌بندی و همچنین رفتار همگرایی و تلاش‌های محاسباتی به‌کارگرفته شده است، چراکه همگرایی سریع در حل مسائل بهینه‌سازی چندهدفه‌ی پرهزینه و مسائلی که فضای جواب بسیار گسترده‌یی دارند، مطلوبیت زیادی به‌همراه دارد. هدف FastPGA یافتن جواب‌های بهینه‌ی پارتویی است که علاوه بر توزیع و گستردگی مناسب در فضای جواب، از نظر محاسباتی معقول و به‌صرفه باشد.

۱.۲.۴. رتبه‌بندی جواب‌ها و تخصیص مقدار شایستگی به جواب‌ها

استراتژی جدید رتبه‌بندی معرفی‌شده در این الگوریتم، جواب‌های کاندید را به دو رتبه‌ی مختلف مبتنی بر رویکرد غلبه تقسیم می‌کند. ابتدا تمامی جواب‌های نامغلوب به‌عنوان جواب‌های رتبه اول و جواب‌های مغلوب به‌عنوان جواب‌های رتبه دوم شناسایی می‌شود؛ سپس مقدار شایستگی جواب‌های رتبه اول از طریق مقایسه‌ی تمام جواب‌های نامغلوب با استفاده از فاصله‌ی ازدحامی تخصیص می‌یابد. هر جواب مغلوب در دسته‌ی دوم با تمام جواب‌های دیگر (شامل جواب‌های غالب و مغلوب) در جمعیت مقایسه می‌شود و یک مقدار شایستگی براساس تعداد جواب‌هایی که مغلوب می‌سازد به آن تعلق می‌گیرد. در اینجا به هر جواب x_i در جمعیت یک مقدار توان خالص[13] $S(x_i)$ تعلق می‌گیرد که نشان‌دهنده‌ی تعداد جواب‌هایی است که مغلوب می‌سازد (رابطه‌ی ۱۵).

$$S(x_i) = |\{x_j | \forall x_j \in CP_t \wedge x_i \succ x_j \wedge j \neq i\}| \qquad (۱۵)$$

کاردینال یک مجموعه با $|.|$ نشان داده می‌شود. CP_t مجموعه جواب‌های نسل t است و عبارت $x_i \succ x_j$ به این معنی است که جواب x_i جواب x_j را مغلوب می‌سازد. سپس مقدار شایستگی هر جواب مغلوب از طریق رابطه‌ی ۱۶ محاسبه می‌شود:

$$F(x_i) = \sum_{x_i \succ x_j} S(x_J) - \sum_{x_k \succ x_i} S(x_k), \quad \forall x_j, x_k \in CP_t \wedge j \neq i \neq k \qquad (۱۶)$$

به عبارت دیگر مقدار شایستگی هر جواب مغلوب x_i برابر است با تفاضل مجموع مقدار توان تمام جواب‌هایی که جواب x_i مغلوب می‌سازد با مجموع تمام جواب‌هایی که بر جواب x_i غالب‌اند. در این رویکرد به‌دلیل عدم استفاده از هیچ سازوکار حفظ تنوع دیگری در میان جواب‌های نامغلوب، به محاسبات کم‌تری نیاز است.

بعد از محاسبه‌ی مقادیر شایستگی تمام جواب‌های جمعیت و مقایسه‌ی آن‌ها سه سناریو اتفاق می‌افتد: ۱. دو جواب با رتبه‌های مختلف مقایسه می‌شوند، در این شرایط جواب با رتبه‌ی بهتر برتری دارد. ۲. دو جواب رتبه‌های برابر اما مقدار شایستگی متفاوت دارند، در این حالت جواب با مقدار شایستگی بالاتر انتخاب می‌شود. ۳. دو جواب دارای رتبه و مقدار شایستگی برابرند، که در این‌صورت یکی از آن‌ها به‌صورت تصادفی انتخاب می‌شود.

۲.۲.۴. نخبه‌گرایی و تعدیل جمعیت

با ترکیب جمعیت نسل قبلی و جمعیت حاصل از تولید فرزندان این فرصت فراهم می‌شود که جواب‌های برتر در نسل‌های بعدی حفظ، و از جواب‌های نامرغوب (جواب‌های با رتبه دو) چشم‌پوشی شود. اگر اندازه جمعیت خیلی بزرگ و در طول نسل‌های مختلف ثابت باشد، به کاهش نخبه‌گرایی در نسل‌های اولیه منجر می‌شود. علاوه بر این وجود عامل نوسان در تعداد جواب‌های نامغلوب در طول نسل‌های مختلف نیازمند یک استراتژی اندازه جمعیت انطباقی برای تأکید بر شدت نخبه‌گرایی روی جواب‌های نامغلوب است. اگر شدت نخبه‌گرایی بسیار بالا باشد، ممکن است همگرایی زودرس اتفاق بیفتد و اگر شدت نخبه‌گرایی بسیار کم باشد، ممکن است همگرایی بسیار دیر اتفاق بیفتد و هزینه‌ی محاسباتی بالایی را تحمیل کند. بنابراین FastPGA یک عملگر تعدیل را برای تنظیم اندازه جمعیت به‌صورت پویا و تا زمانی که اندازه جمعیت به یک مقدار از پیش تعیین‌شده برسد، به‌کار می‌گیرد. عملگر تعدیل از رابطه‌ی ۱۷ استفاده می‌کند:

$$|P_t| = \min\{a_t + \lceil b_t \times |\{x_i | x_i \in CP_t \wedge x_i \text{ isnondominated}\}| \rceil \\ , \text{maxpopsize}\} \qquad (۱۷)$$

که در آن $|P_t|$ اندازه جمعیت در نسل t است، مقادیر a_t و b_t نیز به‌ترتیب مقادیر صحیح مثبت و حقیقی مثبت هستند که ممکن است در طول نسل‌های مختلف تغییرکند، $\lceil x \rceil$ کوچک‌ترین عدد صحیح بزرگ‌تر یا مساوی با مقدار حقیقی x است و maxpopsize بیانگر بیشترین اندازه جمعیت از پیش تعیین‌شده است.

الگوریتم FastPGA برخلاف بسیاری از روش‌های بهینه‌سازی تکاملی دیگر، از مزایای به‌کارگیری اندازه فرزندان تولید شده از طریق عملگرهای تقاطع و جهش به‌صورت پویا بهره می‌جوید. تعداد فرزندان تولید شده در هر نسل از طریق رابطه‌ی ۱۸ محاسبه می‌شود:

$$|O_t| = \min\{c_t + \lceil d_t \times |\{x_i | x_i \in CP_t \wedge x_i \text{ isnondominated}\}| \rceil \\ , \text{maxsoleval}\} \qquad (۱۸)$$

که در آن $|O_t|$ اندازه جمعیت فرزندان تولید شده در نسل t، مقادیر c_t و d_t به‌ترتیب مقادیر صحیح مثبت و حقیقی مثبت و maxsoleval بیانگر بیشینه تعداد ارزیابی جواب‌ها در هر نسل است. این ویژگی پویا بودن، FastPGA را قادر می‌سازد تا در تعداد قابل توجه ارزیابی جواب‌ها در ابتدای جست‌وجو صرفه‌جویی به عمل آورد و استخراج[14] را در یک رفتار کارا در نسل‌های بعد به‌کار گیرد.

۳.۴. عملگرهای مورد استفاده در دو الگوریتم

۱.۳.۴. نحوه‌ی نمایش جواب

در مسائل موازنه‌ی زمان ـ هزینه ـ کیفیت برای هر فعالیت چندین حالت اجرا وجود دارد که باید فقط یک حالت اجرا از بین آن‌ها انتخاب شود. لذا در اینجا به نوع نمایشی نیاز داریم که برای هر فعالیت، یک حالت اجرا را از بین حالت‌های اجرای ممکن و مجاز نمایش دهد. مناسب‌ترین نوع نمایش برای این دسته از مسائل نمایش عدد صحیح است به‌طوری که یک کروموزوم مجموعه‌یی از مقادیر عدد صحیح (ژن‌ها) است که نشان‌دهنده‌ی حالت‌های اجرای انتخابی برای فعالیت‌های یک مسئله است. در شکل ۳ یک نمایش از یک جواب‌شدنی برای مسئله‌یی با ۱۰ فعالیت نشان داده شده است. برای نمونه، عدد ۳ در موقعیت ژن دوم به‌معنی انتخاب حالت اجرای ۳ برای فعالیت دوم است.

۲.۳.۴. محاسبه‌ی توابع هدف

مقایسه‌ی جواب‌های مختلف با یکدیگر و رتبه‌بندی آن‌ها براساس محاسبه‌ی مقادیر توابع هدف صورت می‌گیرد. در واقع براساس قانون غلبه، هنگامی یک جواب بر جواب دیگر ارجحیت دارد که در هیچ‌کدام از توابع مربوط به زمان، هزینه و کیفیت از جواب دیگر بدتر نباشد و دست کم در یکی از این اهداف از جواب دیگر بهتر باشد.

۳.۳.۴. عملگر تقاطع

عملگر تقاطع وابسته به نوع نمایش است و برای نمایش‌های مختلف، عملگرهای متفاوتی باید تعریف شود. در اینجا با توجه به نوع نمایش، از تقاطع یکنواخت[15]

۴	۳	۱	۲	۴	۳	۵	۴	۲	۱

شکل ۳. مثالی از نحوه‌ی نمایش مسئله‌ی مدل شده.

ــ که از جمله عملگرهای تقاطع مربوط به نمایش عدد صحیح است ــ استفاده شده است. در تقاطع یکنواخت، دنباله‌یی از متغیرهای تصادفی از توزیع یکنواخت در بازه [۰ , ۱] و به تعداد ژن‌های کروموزوم تولید می‌شود. در هر موقعیت اگر مقدار، کم‌تر از پارامتر P_c باشد، ژن از والد اول و در غیر این‌صورت از والد دوم به ارث برده می‌شود. فرزند دوم با فرایندی عکس این فرایند تولید می‌شود (شکل ۴).

۴.۳.۴. عملگر جهش

همانند عملگر تقاطع، نوع عملگر مورد استفاده برای جهش نیز به نوع نمایش بستگی دارد. در اینجا ما از عملگر جهش جایگزینی تصادفی[16] ــ که از روش‌های رایج برای نمایش‌های عدد صحیح است ــ استفاده کرده‌ایم. در این عملگر ابتدا یک رشته از اعداد تصادفی به‌طول تعداد ژن‌ها در بازه [۰ , ۱] تولید می‌شود، در هر موقعیت از رشته اگر عدد تصادفی مربوطه از مقدار نرخ جهش کم‌تر باشد یک مقدار جدید به‌طور تصادفی از مجموعه‌ی مقادیر مجاز در هر موقعیت، انتخاب شده و جایگزین مقدار ژن در آن موقعیت می‌شود. در غیر این‌صورت مقدار ژن بدون تغییر باقی می‌ماند. در شکل ۵، جهش برای ژن‌های ۲، ۵ و ۷ اتفاق افتاده است.

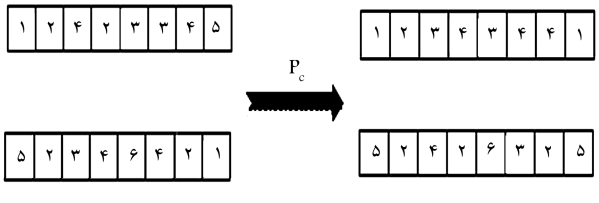
۵.۳.۴. سازوکار انتخاب والد

در ساختار کلی الگوریتم دو مرحله در چرخه‌ی تکامل وجود دارد که در آن‌ها رقابت مبتنی بر برازندگی رخ می‌دهد: ۱. مرحله‌ی انتخاب افراد برای شرکت در تولید مثل (انتخاب والد)؛ ۲. مرحله‌ی انتخاب افراد برای بقا در نسل بعدی (انتخاب بازمانده).

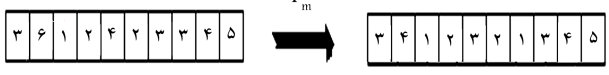
در الگوریتم ژنتیک افراد با برازندگی بالاتر شانس بیشتری برای والد شدن نسبت به افراد با برازندگی پایین‌تر دارند. در اینجا برای انتخاب والدین از مسابقه‌ی باینری[17] استفاده شده است. در این نوع انتخاب، در هر مرحله ۲ جواب انتخاب، و از میان آن‌ها بهترین فرد از لحاظ برازندگی به‌عنوان والد انتخاب می‌شود. این کار تا انتخاب تعداد والدین لازم تکرار می‌شود.

۶.۳.۴. تولید جمعیت اولیه

هر ژن با تخصیص یک حالت اجرا از بین مجموعه‌ی حالت‌های اجرای ممکن برای هر فعالیت به‌صورت تصادفی تشکیل می‌شود. این کار به تعداد جمعیت انجام می‌شود تا جمعیت اولیه به دست آید. منظور از حالت‌های اجرای ممکن برای یک فعالیت، مجموعه حالت‌هایی است که از کیفیت لازم برای انجام آن فعالیت برخوردار است. به‌همین منظور، قبل از تولید جمعیت اولیه، با استفاده از اجرای یک تابع پیش‌پردازش از ورود حالت‌های اجرایی ناکارآمد که مجموع وزنی کیفیت شاخص‌های مربوط به آن از حد معینی پایین‌تر باشد به هر مرحله از تولید جمعیت ممانعت به عمل آمده است. لذا همواره تمامی جواب‌هایی که تولید می‌شوند، جواب‌های شدنی هستند.



شکل ۴. مثالی از فرایند تقاطع یکنواخت.



شکل ۵. مثالی از جهش جایگزینی تصادفی.

۵. نتایج محاسباتی

۱.۵. تولید مثال‌های نمونه

به منظور بررسی و آزمایش عملکرد الگوریتم‌های ارائه شده باید از تعدادی از مثال‌های معیار استفاده کرد. از آنجا که در پیشینه تحقیق هیچ‌گونه مثالی برای مسئله‌ی مورد بحث نیست، لذا در ابتدا به تولید چندین مثال برای مسئله‌ی مطروحه خواهیم پرداخت. تولید مثال‌های مربوط به این بحث در کل به دو بخش تولید داده‌های مربوط به پارامترهای اساسی و تولید شبکه مربوطه تقسیم می‌شود.

۱.۱.۵. تولید داده‌های مربوط به پارامترهای اساسی

تولید داده‌های مربوط به پارامترها براساس یک فرایند تصادفی و با رعایت منطق بین آن‌ها انجام گرفته است. برای نمونه در تولید حالت‌های مختلف برای اجرای فعالیت‌ها حالتی یافت نمی‌شود که در آن به‌طور همزمان هزینه و زمان مربوط به یک حالت اجرا از هزینه و زمان مربوط به حالت اجرای دیگر کم‌تر و کیفیت آن از کیفیت دیگری بیشتر باشد. در اینجا مثال‌هایی با ابعاد مختلف تولید شده که در ادامه به‌اختصار نحوه‌ی تولید مثال ششم بیان شده است. در این مثال، به هر فعالیت به‌صورت تصادفی وزنی از مجموعه‌ی:

$$\{\frac{۱}{۱۶۰}, \frac{۲}{۱۶۰}, \frac{۳}{۱۶۰}, \frac{۴}{۱۶۰}, \frac{۵}{۱۶۰}, \frac{۶}{۱۶۰}, \frac{۷}{۱۶۰}, \frac{۸}{۱۶۰}, \frac{۹}{۱۶۰}, \frac{۱۰}{۱۶۰}\}$$

تخصیص یافته است، با این محدودیت که مجموع اوزان باید برابر ۱ شود. تعداد حالت‌های اجرای فعالیت‌ها به‌صورت تصادفی از توزیع یکنواخت گسسته در بازه [۲, ۵] انتخاب شده‌اند. هرکدام از حالت‌های اجرا شامل سه پارامتر زمان، هزینه و کیفیت هستند. زمان انجام هر حالت اجرای فعالیت به‌طور تصادفی از توزیع یکنواخت گسسته در بازه [۱۰, ۶۰] نمونه‌گیری شده است. سپس مدت زمان‌های اجرا به‌صورت نزولی برای هر فعالیت مرتب می‌شود. هزینه‌ی متناظر با حالت اجرای فعالیت‌ها با طولانی‌ترین زمان اجرای آنها به‌صورت تصادفی از توزیع یکنواخت گسسته در بازه [۵۰, ۱۲۰] انتخاب شده‌اند. هزینه‌ی دیگر حالت‌های اجرا بدین صورت تعیین می‌شود که اگر c_k هزینه‌ی انجام یک فعالیت در حالت اجرای k باشد و t_k نیز زمان اجرای آن فعالیت در حالت k باشد آنگاه برای هر فعالیت هزینه‌ی اجرای آن در حالت k به‌صورت تصادفی و از توزیع یکنواخت گسسته در بازه $[c_{k-۱} + (t_k - t_{k-۱}), c_{k-۱} + ۴(t_k - t_{k-۱})]$ انتخاب می‌شود.

برای کیفیت انجام پروژه چهار شاخص لحاظ شده که میزان اهمیت هرکدام از آن‌ها برای فعالیت‌های مختلف متفاوت است. این اوزان به‌صورت تصادفی به‌گونه‌یی به شاخص‌های کیفیت تخصیص داده شده که مجموع آن‌ها برای هر فعالیت برابر با ۱ شود. فرض می‌شود که کاهش زمان اجرای یک فعالیت الزاماً باعث کاهش کیفیت آن نمی‌شود، لذا مقادیر مربوط به شاخص‌های کیفیت انجام هر فعالیت در هر حالت اجرا به‌صورت تصادفی از توزیع یکنواخت گسسته در بازه [۶۰, ۹۹] تولید شده است.

۲.۱.۵. تولید شبکه

ساختار کلی پروژه را با استفاده از یک شبکه گرهی نمایش می‌دهیم که در آن گره‌ها بیان‌گر فعالیت‌ها، و کمان‌ها نشان‌دهنده‌ی روابط پیش‌نیازی بین فعالیت‌ها هستند. سابقاً تصور می‌شد که هرقدر تعداد کمان‌ها بیشتر باشد، شبکه‌ی مربوطه پیچیده‌تر است (پیچیدگی[18] یک شبکه برابر است با نسبت تعداد کمان‌های غیر زائد به تعداد گره‌های آن)، اما محققین خلاف این را بیان کردند[22] چرا که ممکن است تعداد زیادی از این کمان‌ها زائد و اضافی باشد. آن‌ها چهار حالت مختلف را برای زائد بودن کمان ارائه کردند؛ در مثال‌های تولید شده سعی بر آن بوده که از تولید کمان‌های زائد

جدول ۱. پارامترهای مربوط به مثال‌های تولید شده.

شماره مسئله	تعداد فعالیت‌ها	تعداد حالت‌های اجرا	زمان	هزینه	تعداد شاخص‌های کیفیت	کیفیت هر شاخص در هر حالت اجرا	حداقل مجموع شاخص‌های کیفیت در هر حالت اجرا	C
۱	۷	[۲, ۹]	[۵, ۳۰]	[۱۲۰, ۱۹۰]	۲	[۶۰, ۹۹]	[۶۰, ۷۵]	۱/۴۳
۲	۹	[۲, ۴]	[۱, ۱۵]	[۱۰, ۲۰]	۵	[۶۰, ۹۹]	[۶۰, ۷۵]	۱/۵۶
۳	۱۵	[۲, ۷]	[۱, ۲۰]	[۳۰, ۶۰]	۲	[۶۰, ۹۹]	[۶۰, ۷۵]	۲/۰۶
۴	۲۱	[۲, ۶]	[۵, ۴۰]	[۲۵۰, ۴۸۰]	۳	[۶۰, ۹۹]	[۶۰, ۷۵]	۲/۲۹
۵	۳۱	[۲, ۶]	[۱۵, ۵۰]	[۵۰, ۱۴۰]	۴	[۶۰, ۹۹]	[۶۰, ۷۵]	۲
۶	۴۰	[۲, ۵]	[۱۰, ۶۰]	[۵۰, ۱۲۰]	۴	[۶۰, ۹۹]	[۶۰, ۷۵]	۱/۹
۷	۵۰	[۲, ۵]	[۳۰, ۸۰]	[۸۰, ۱۸۰]	۳	[۶۰, ۹۹]	[۶۰, ۷۵]	۱/۸

خودداری شود. از طرفی روابط بین فعالیت‌ها از نوع روابط پیش‌نیازی تعمیم‌یافته است که در آن‌ها با توجه به این که روابط از نوع بیشینه قابلیت تبدیل به روابط از نوع کمینه و در جهت مخالف را دارند، در تولید شبکه‌های مربوطه فقط روابط پیش‌نیازی از نوع کمینه‌لحاظ شده است. در جدول ۱ پارامترهای مربوط به مثال‌های تولیدشده ثبت شده است.

۲.۵. تنظیم پارامترها

الگوریتم‌های فرااِبتکاری نسبت به پارامترهای خود بسیار حساس‌اند و تغییر پارامترها تأثیر به‌سزایی بر کارایی و اثربخشی جست‌وجو دارند. الگوریتم‌های پیشنهادی در نوشتار حاضر با استفاده از زبان برنامه‌نویسی Matlab کد شده و در سیستم رایانه با مشخصات پردازش‌گر Core i7 ۲/۹۳ GHZ و حافظه‌ی جانبی ۶ GB اجرا شده است. در جدول ۲ مقادیر پیشنهادی برای هریک از پارامترهای مربوط به دو

جدول ۲. مقادیر پیشنهادی برای تنظیم پارامترها.

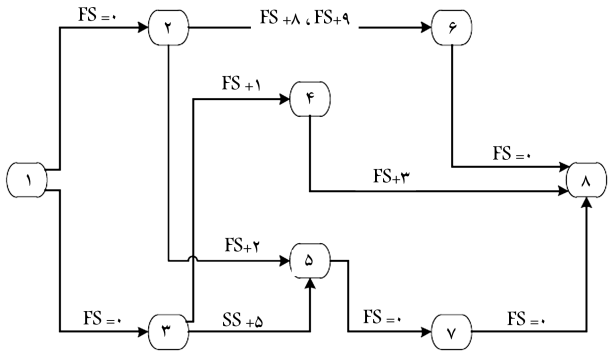
نوع الگوریتم	پارامتر	مقادیر پیشنهادی		
NSGA-II	نرخ جهش	۰/۱	۰/۲	۰/۳
	نرخ تقاطع	۰/۴	۰/۵	۰/۶
	تعداد جمعیت هر نسل	$5n$	$7n$	$10n$
FastPGA	a	$2n$	$3n$	$4n$
	b	۰/۵	۱	۱/۵
	c	$2n$	$3n$	$4n$
	d	۰	۰/۵	۱
	نرخ جهش	۰/۱	۰/۲	۰/۳
	نرخ تقاطع	۰/۴	۰/۵	۰/۶
	بیشینه تعداد جمعیت	$5n$	$7n$	$10n$
	بیشینه تعداد جمعیت فرزندان	$5n$	$7n$	$10n$

روش NSGA-II و FastPGA بیان شده است. لازم به ذکر است که در این جدول n بیان‌گر تعداد فعالیت‌های مربوط هر مسئله است.

آزمایشات روی مسائل شماره ۲ و شماره ۶ تولید شده به‌منظور بررسی و تنظیم پارامترها برای مسائل با ابعاد کوچک (مسائل شماره ۱، ۲، ۳ و ۴) و مسائل با ابعاد متوسط و نسبتاً بزرگ (مسائل شماره ۵، ۶ و ۷) انجام گرفته است. در جداول ۳ و ۴ مقادیر تنظیم پارامتر برای دو روش ارائه شده که در آن‌ها مقدار هر پارامتر با ثابت نگه‌داشتن سایر پارامترها درکم‌ترین مقدار خود، براساس تعداد نقاط نامغلوب شناسایی شده در تعداد مشخصی تکرار، تنظیم شده است.

۳.۵. اعتبارسنجی الگوریتم‌های پیشنهادی

به‌منظور حصول اطمینان از این که نتایج حاصل از مدل کاملاً به سمت جبهه‌ی پارتویی واقعی مسئله نیل می‌کند و پاسخ‌ها از پراکندگی لازم برخوردارند، باید میزان کارایی روش‌های مورد استفاده در اجزای الگوریتم‌ها و همچنین اعتبار جواب‌های به دست آمده مورد سنجش قرار گیرد. به‌همین منظور سه مثال در ابعاد نسبتاً کوچک تولید شده و با جست‌وجوی تمام فضای ممکن در این مسائل جبهه‌ی واقعی آن‌ها شناسایی شده و با نتایج حاصل از اجرای الگوریتم‌ها مقایسه می‌شوند. در شکل ۶ شبکه‌ی مربوط به اولین مثال، و در جدول ۵ مشخصات مثال‌های تولیدی و نتایج مربوط به جست‌وجوی کامل فضای جواب و همچنین حل آن‌ها با استفاده از دو



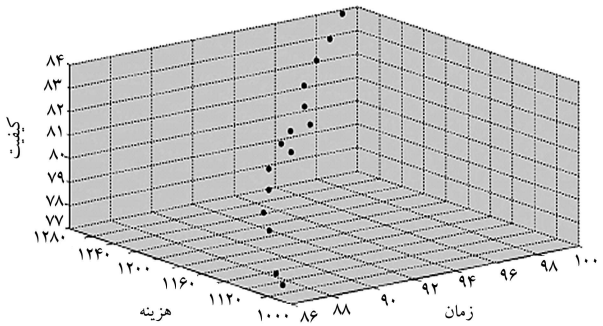
شکل ۶. شبکه مثال تولیدی اول جهت ارزیابی الگوریتم‌های حل.

جدول ۳. مقادیر تنظیم پارامتر برای پارامترهای الگوریتم FastPGA.

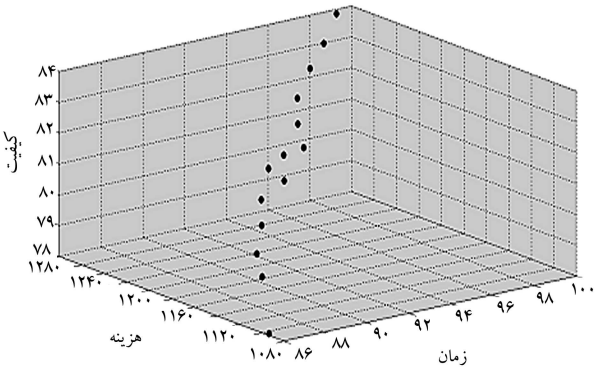
	a	b	c	d	نرخ جهش	نرخ تقاطع	بیشینه تعداد جمعیت	بیشینه تعداد جمعیت فرزندان
مثال‌های کوچک	$4n$	۰/۵	$3n$	۰/۵	۰/۱	۰/۵	$7n$	$5n$
مثال‌های متوسط و نسبتاً بزرگ	$3n$	۰/۵	$3n$	۱	۰/۱	۰/۴	$7n$	$5n$

جدول ۴. مقادیر تنظیم پارامتر برای پارامترهای الگوریتم NSGA-II.

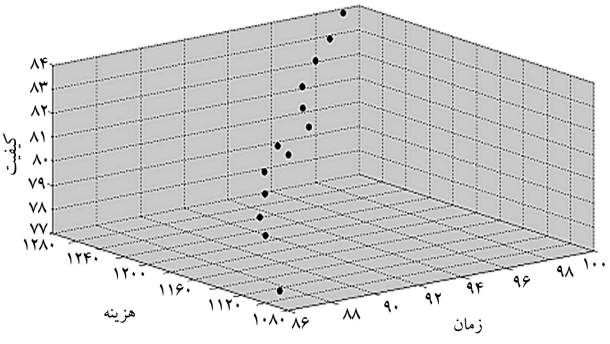
	نرخ جهش	نرخ تقاطع	تعداد جمعیت هر نسل
مثال‌های کوچک	۰/۲	۰/۴	$10n$
مثال‌های متوسط و نسبتاً بزرگ	۰/۳	۰/۵	$7n$



الف) پراکندگی نقاط پارتو در فضای جواب با استفاده از روش شمارش کامل؛



ب) پراکندگی نقاط نامغلوب شناسایی شده در الگوریتم NSGA-II ؛



ج) پراکندگی نقاط نامغلوب شناسایی شده در الگوریتم FastPGA .

شکل ۷. پراکندگی نقاط نامغلوب.

الگوریتم نشان داده شده است. برای نمونه پس از بررسی کامل فضای جواب مثال اول مشخص شد که تعداد ۱۵ نقطه‌ی نامغلوب برای این مثال وجود دارد (جدول ۶). چنان که مشاهده می‌شود این جواب‌ها هیچ‌گونه برتری نسبت به یکدیگر ندارند و جوابی یافت نمی‌شود که در تمامی اهداف از جواب دیگر بهتر باشد. پراکندگی این نقاط در فضای جواب در شکل ۷الف نشان داده شده است. مسئله‌ی فوق توسط دو الگوریتم NSGA-II و FastPGA و با مد نظر قرار دادن پارامترهای تعیین شده در بخش تنظیم پارامتر حل شد. الگوریتم‌های NSGA-II و FastPGA در مدت حدوداً ۴ ثانیه توانستند به ترتیب ۱۴ و ۱۳ نقطه از نقاط نامغلوب را شناسایی کنند. با مشاهده‌ی نتایج حاصل از دو مثال دیگر می‌توان نتیجه گرفت که صحت و توانایی دو الگوریتم در رسیدن به جواب‌های بهینه قابل قبول است. در شکل‌های ۷ب و ۷ج نقاط نامغلوب شناسایی شده در دو الگوریتم نمایش داده شده است.

۴.۵. معیارهای ارزیابی و مقایسه‌ی الگوریتم‌ها

به‌منظور مقایسه‌ی کارایی دو الگوریتم، از چندین شاخص کاربردی برای مقایسه‌ی الگوریتم‌های چندهدفه استفاده خواهیم کرد. این شاخص‌ها عمدتاً به دو دسته تقسیم می‌شوند: دسته‌ی اول بر همگرایی و کیفیت جواب‌ها تأکید دارد و دسته‌ی دوم پراکندگی و گسترش جواب‌ها را در فضای حل مورد توجه قرار می‌دهد. در اینجا از چهار شاخص برای مقایسه‌ی دو الگوریتم استفاده شده است:

۱. تعداد جواب‌های پارتو (NOS)[۱۹]؛[۲۳]

۲. بیشترین گسترش (MS)[۲۰]؛[۲۴]

۳. میانگین فاصله از جواب ایده‌آل (MID)؛[۲۵]

۴. تنوع (DM)[۲۱].[۲۶]

۵.۵. نتایج محاسباتی

در این بخش به حل مثال‌های تولید شده در بخش ۱.۵ با دو الگوریتم NSGA-II و FastPGA می‌پردازیم. به‌منظور ارزیابی و مقایسه‌ی دو الگوریتم، از معیارهای بخش ۴.۵ استفاده می‌شود. هر مثال توسط هر الگوریتم ۱۰بار اجرا شده و جواب‌های نامغلوب این ۱۰ اجرا برای مقایسه‌ی نتایج در نظرگرفته می‌شود. مقایسه‌ی الگوریتم‌ها در دو بخش مثال‌های کوچک (مثال‌های ۱ تا ۴) و مثال‌های متوسط و بزرگ (مثال‌های ۵ تا ۷) ارائه شده است.

۱.۵.۵. ارزیابی عملکرد الگوریتم‌های پیشنهادی در مسائل کوچک

در این قسمت عملکرد و کارایی دو الگوریتم NSGA-II و FastPGA برای مثال‌های ۱، ۲، ۳ و ۴ مورد سنجش و مقایسه قرار می‌گیرد. در جدول ۷ بهترین جواب‌های کسب‌شده (ترکیب‌های سه‌تایی از زمان، هزینه و کیفیت) در ۱۰ مرتبه اجرا و همچنین میانگین جواب‌های به دست آمده از این تعداد اجرا برای دو الگوریتم مورد استفاده، ارائه شده است. همچنین در جدول ۸ مقادیر مربوط به شاخص‌های مقایسه‌یی برای هر دو الگوریتم ارائه شده است. زمان اجرا برای دو الگوریتم یکسان

جدول ۵. مشخصات و نتایج مربوط به مثال‌های تولید شده جهت اعتبارسنجی الگوریتم‌های حل.

زمان اجرای دو الگوریتم (ثانیه)	تعداد نقاط نامغلوب کشف شده توسط الگوریتم		کل زمان حل (ثانیه)	تعداد کل نقاط نامغلوب	تعداد کل نقاط شدنی فضای جواب	ضریب پیچدگی	تعداد مدها	تعداد فعالیت‌ها
	FastPGA	NSGA-II						
۴	۱۳	۱۴	۲۸۷	۱۵	۵۷۶۰	۱٫۳۸	[۲, ۵]	۸
۳۰	۲۸	۳۱	۲۳۷۰	۳۷	۴۰۵۰۰	۱٫۵	[۲, ۵]	۹
۳۵	۱۱	۱۲	۲۰۷۲۰	۱۲	۱۸۶۶۲۴	۱٫۴۲	[۴, ۲]	۱۲

جدول ۶. مقادیر توابع هدف مربوط به نقاط نامغلوب مثال اول.

نقاط نامغلوب	زمان	هزینه	کیفیت
۱	۹۹	۱۲۷۵	۸۳٫۹۶
۲	۹۷	۱۲۴۹	۸۳٫۶۶
۳	۹۵	۱۲۲۴	۸۳٫۴۸
۴	۹۰	۱۱۶۳	۸۱٫۸
۵	۹۳	۱۱۹۸	۸۳٫۱۸
۶	۹۱	۱۱۷۳	۸۲
۷	۸۸	۱۱۳۷	۸۱٫۵
۸	۹۲	۱۱۷۹	۸۲٫۷۶
۹	۹۰	۱۱۵۴	۸۱٫۵۷
۱۰	۸۷	۱۱۱۸	۸۱٫۰۷
۱۱	۸۶	۱۱۰۴	۸۰٫۵۱
۱۲	۹۲	۱۱۷۴	۸۲٫۰۷
۱۳	۸۶	۱۰۹۹	۷۹٫۸۲
۱۴	۸۶	۱۰۹۳	۷۸٫۰۷
۱۵	۸۶	۱۰۸۷	۷۷٫۷

در نظرگرفته شده، لذا از مقایسه‌ی زمان‌های اجرا صرف‌نظر شده است و تنها مقادیر آن‌ها در مثال‌های مختلف در ستون دوم نشان داده شده است.

از جدول ۸ می‌توان دریافت که الگوریتم FastPGA در یافتن تعداد نقاط پارتویی در تمامی مسائل به‌جز مسئله‌ی شماره ۴ موفق‌تر از الگوریتم NSGA-II عمل می‌کند. در ارتباط با میانگین فاصله از جواب ایده‌آل نمی‌توان درمورد برتری یک الگوریتم نسبت به دیگری اظهار نظر قطعی کرد. برای معیارهای بیشترین گسترش و تنوع که دلالت بر پراکندگی و توزیع مناسب جواب‌ها دارند برتری الگوریتم FastPGA بر NSGA-II محسوس است.

۲.۵.۵. ارزیابی عملکرد الگوریتم‌های پیشنهادی در مسائل متوسط و بزرگ

در این قسمت مانند بخش قبلی کارایی الگوریتم‌های پیشنهادی در ارتباط با سه مسئله‌ی ۵، ۶ و ۷ مورد بررسی قرار می‌گیرد. در جدول ۹ بهترین جواب‌های کسب شده و همچنین میانگین جواب‌های به دست آمده برای دو الگوریتم ارائه شده است. مقادیر مربوط به شاخص‌های مقایسه‌یی برای دو الگوریتم نیز در جدول ۱۰ نمایش داده شده است.

با توجه به نتایج ارائه شده در جدول ۱۰ می‌توان دریافت که روش NSGA-II در ارتباط با کیفیت و تعداد جواب‌های نامغلوب نتایج بهتری را از خود نشان داده در حالی که این امر در روش FastPGA در ارتباط با گسترش جواب‌ها صادق است. مطابق انتظار، با افزایش ابعاد مسائل و بزرگ‌تر شدن فضای جواب روش NSGA-II در مقایسه‌با روش FastPGA تعداد نقاط پارتویی (نزدیک به مرز پارتو)

جدول ۷. نتایج محاسباتی دو الگوریتم برای مثال‌های ۱ تا ۴.

مسئله	نوع الگوریتم	بهترین جواب‌های به دست‌آمده			میانگین جواب‌های به دست‌آمده		
		زمان	هزینه	کیفیت	زمان	هزینه	کیفیت
۱	NSGA-II	۷۳٫۰۹	۱۹۳۶٫۵۹	۸۵٫۴۷	۷۶٫۸۲	۱۹۵۱٫۳۷	۸۳٫۰۳
	Fast-PGA	۷۳٫۴۳	۱۹۳۷٫۸۲	۸۳٫۳۳	۷۵٫۵۳	۱۹۴۷٫۷۳	۸۲٫۳۱
۲	NSGA-II	۴۰٫۴۴	۶۰۰٫۱	۹۱٫۳	۴۲٫۳۴	۶۱۰٫۸۸	۸۸٫۶۸
	Fast-PGA	۴۱٫۷۳	۶۰۷	۸۹٫۷۹	۴۲٫۷۴	۶۱۴٫۵۵	۸۸٫۹۴
۳	NSGA-II	۵۷٫۶	۱۳۳۹٫۵۲	۸۵٫۲۶۲	۵۸٫۹۹	۱۳۴۸٫۶۸	۸۴٫۷۹
	Fast-PGA	۵۷٫۱۸۳	۱۳۳۶٫۳۳	۸۴٫۸۹	۵۸٫۵۷	۱۳۴۵٫۱۸	۸۴٫۶۸
۴	NSGA-II	۱۷۱٫۴۰۵	۱۰۲۵۳٫۲۱	۸۶٫۹۶	۱۷۲٫۲۷	۱۰۳۱۶٫۴۱	۸۶٫۴۶
	Fast-PGA	۱۷۱٫۳۷	۱۰۲۸۸٫۱۳	۸۶٫۶۲	۱۷۲٫۸۷	۱۰۳۱۰٫۱۵	۸۶٫۲۹

جدول ۸. معیارهای مقایسه‌ی دو الگوریتم NSGA-II و FastPGA برای مثال‌های ۱ تا ۴.

مسئله	نوع الگوریتم	زمان اجرا (ثانیه)	تعداد جواب‌های پارتو	بیشترین گسترش	میانگین فاصله از جواب ایده‌آل	تنوع
۱	NSGA-II	۱۰	۲۹	۱۱۳٫۵۲	۵۰٫۴۶	۱۲٫۳۲
	Fast-PGA	۱۰	۳۰	۱۱۴٫۲۲	۴۴٫۷۶	۱۲٫۵
۲	NSGA-II	۱۰	۲۳	۷۷٫۷۲	۲۷٫۳۴	۱۰٫۰۶
	Fast-PGA	۱۰	۲۴	۸۰٫۹۳	۳۰٫۰۸	۱۰٫۲۴
۳	NSGA-II	۳۰	۱۷	۶۳٫۲۹	۳۱٫۴۶	۸٫۹۴
	Fast-PGA	۳۰	۲۵	۶۵٫۲۱	۲۷٫۷۹	۹٫۰۲
۴	NSGA-II	۱۸۰	۱۲۲	۴۷۱٫۱۲	۲۰۲٫۷	۲۲٫۹۶
	Fast-PGA	۱۸۰	۱۱۸	۵۴۰٫۱۹	۲۱۷٫۶۲	۲۴٫۰۱

جدول ۹. نتایج محاسباتی دو الگوریتم برای مثال‌های ۵ تا ۷.

مسئله	نوع الگوریتم	بهترین جواب‌های به دست‌آمده			میانگین جواب‌های به دست‌آمده		
		زمان	هزینه	کیفیت	زمان	هزینه	کیفیت
۵	NSGA-II	۲۸۳	۶۱۲۸٫۷	۸۷٫۵۴	۲۸۶٫۵۶	۶۱۹۵٫۸	۸۷٫۱۳
	Fast-PGA	۲۸۹٫۷۱	۶۲۰۹٫۵	۸۷٫۳۴	۲۹۱٫۷	۶۲۴۰٫۵	۸۷٫۰۸
۶	NSGA-II	۴۱۶٫۳۸	۸۰۵۶٫۹	۸۲٫۷۸	۴۲۱٫۰۱	۸۱۰۰٫۳	۸۲٫۶۵
	Fast-PGA	۴۲۶٫۵۴	۸۱۴۱٫۴	۸۲٫۷۳	۴۳۰٫۸۲	۸۱۹۸٫۲	۸۲٫۶۶
۷	NSGA-II	۴۲۱٫۳۶	۸۱۱۰٫۷	۸۵٫۹۴	۵۶۸٫۹۸	۱۲۴۷۷	۸۵٫۳۲
	Fast-PGA	۲۲٫۷۸۲	۱۳۴۴۶	۸۵٫۶۵	۶۰۶٫۷۱	۱۳۴۹۸	۸۵٫۵۱

جدول ۱۰. معیارهای مقایسه‌ی دو الگوریتم NSGA-II و FastPGA برای مثال‌های ۵ تا ۷.

مسئله	نوع الگوریتم	زمان اجرا (ثانیه)	تعداد جواب‌های پارتو	بیشترین گسترش	میانگین فاصله از جواب ایده‌آل	تنوع
۵	NSGA-II	۶۰۰	۵۳	۳۷۸٫۶	۱۱۴٫۱۸	۲۰٫۰۵
	Fast-PGA	۶۰۰	۵۱	۵۴۵٫۸۲	۱۵۹٫۰۹	۲۴٫۳
۶	NSGA-II	۹۰۰	۱۰۴	۵۲۵٫۰۶	۱۸۳٫۸۹	۲۳٫۸۱
	Fast-PGA	۹۰۰	۵۳	۷۶۱٫۵۸	۲۷۱٫۳۲	۲۹٫۱۲
۷	NSGA-II	۱۰۰۰	۱۱۴	۴۹۳٫۱۳	۲۰۴٫۰۹	۲۳٫۳۵
	Fast-PGA	۱۰۰۰	۴۳	۷۲۴٫۶	۲۴۹٫۹۵	۲۸٫۰۷

بیشتری را می‌یابد، که علت آن همگرایی سریع روش FastPGA در دست‌یابی به نقاط نامغلوب است و تکرارهای بالاتر در این روش منجر به تولید جواب‌های تکراری می‌شود. چنان که مشاهده شد در مورد مثال‌های با ابعاد کوچک، دو الگوریتم تقریباً عملکرد برابری را از خود به نمایش گذاشتند و اختلاف موجود در نتایج آن‌ها قابل چشم‌پوشی است، اما با افزایش ابعاد مسئله، رفته‌رفته این تفاوت‌ها بیشتر بروز یافت. البته نمی‌توان نقش مدت زمان اجرا را در دست‌یابی به جواب‌های خوب نادیده گرفت. واضح است که هرقدر زمان اجرا را افزایش دهیم در واقع به سود روش NSGA-II عمل می‌کنیم و در صورت کاهش مدت اجرا گردونه‌ی رقابت به‌نفع روش FastPGA خواهد چرخید. البته چنان که عنوان شد، با توجه به همگرایی بالای الگوریتم، روش FastPGA در مواردی کاربرد دارد که ارزیابی جواب‌ها از نظر محاسباتی یا مالی هزینه‌بر است، یا به‌دنبال یافتن جواب‌های خوب در یک مدت زمان کوتاه هستیم.

۶. نتیجه‌گیری

با توجه به مطالعات انجام شده می‌توان به اهمیت بسیار زیاد سه عامل زمان، هزینه و کیفیت در قراردادهای امروزی پی برد. با در نظرگرفتن هر سه عامل به‌طور

همزمان در مدل‌های برنامه‌ریزی، و همچنین با نزدیک‌سازی محدودیت‌های مسئله به محدودیت‌های موجود در عالم واقعیت، بدون تردید می‌توان نتایج سودمند و جواب‌های دقیق‌تری دریافت کرد. به‌همین منظور در این مقاله مسئله‌ی تبادل زمان ـ هزینه ـ کیفیت درحالت گسسته و با محدودیت‌های پیش‌نیازی تعمیم‌یافته مورد مطالعه قرار گرفته است، که در آن برای هر فعالیت چندین شیوه اجرا تعریف شده و هر شیوه‌ی اجرا نیز دارای زمان، هزینه و کیفیت متفاوتی است. به‌دلیل این که مسئله‌ی موازنه‌ی زمان ـ هزینه ـ کیفیت در حالت گسسته جزء مسائل NP-hard بوده و با افزایش ابعاد مسئله، زمان حل آن به‌صورت نمایی افزایش می‌یابد، از الگوریتم‌های فراابتکاری NSGA-II و FastPGA که ازجمله الگوریتم‌های مورد استفاده در مسائل چندهدفه هستند، استفاده کردیم. پس از تنظیم پارامترهای مربوطه، به مقایسه‌ی دو الگوریتم برای چندین مثال تولیدشده در ابعاد مختلف و با استفاده از چندین شاخص مقایسه‌یی پرداختیم. در نهایت نتایج محاسباتی و تحلیل آن‌ها برای مثال‌های مختلف ارائه شد.

به‌عنوان توسعه‌یی برای پژوهش‌های آتی می‌توان مدل ارائه شده در این تحقیق را در یک مطالعه موردی واقعی پیاده‌سازی کرد و به بررسی میزان کارایی مدل و روش‌های حل ارائه شده پرداخت. از طرفی ممکن است هرکدام از عوامل زمان، هزینه و کیفیت با عدم قطعیت همراه باشند که در این‌صورت استفاده از منطق فازی به‌عنوان یک ابزار قوی در بررسی عدم قطعیت‌ها می‌تواند مورد توجه قرار گیرد.

پانوشت‌ها

1. non-dominated sorting genetic algorithm (NSGA-II)
2. fast Pareto genetic algorithm
3. critical path method
4. crash activity time
5. lag
6. discrete time-cost trade-off problem
7. the multi-mode problem with generalized precedence relations
8. time-switch constraints
9. discrete time-cost-quality trade-off problem
10. front
11. crowding distance
12. diversity
13. net strength value
14. exploitation
15. uniform crossover
16. random resetting
17. Binary tournament selection
18. complexity
19. number of Pareto solution
20. maximum spread or diversity
21. diversification metric

منابع (References)

1. Kelly, J.E. and Walker, M.R. "Critical-path planning and scheduling: Mathematical basis", *European Journal of Operational Research,* **9**(3), pp. 296-320 (1961).
2. Fulkerson D. "A network flow computation for project cost curves", *Management Science*, **7**, pp. 167-178 (1961).
3. Moder, J.J., Phillips, C.R. and Davis, E.W., *Project Management with CPM*, PERT and Precedence Diagramming (3rd ed.) (1983).
4. Falk, J. and Horowitz, J. "Critical path problem with concave cost curves", *Management Science*, **19**, pp. 446-455 (1972).
5. Hindelang, T.J. and Muth, J.F. "A dynamic programming algorithm for decision CPM networks", *European Journal of Operational Research*, **27**(2), pp. 225-241 (1979).
6. Prabuddha, D., Dunne, E.J., Ghosh, J.B. and Wells, C.E. "Complexity of the discrete time–cost tradeoff problem for project networks", *European Journal of Operational Research*, **45**(2), pp. 302-306 (1997).
7. Liu, S.X., Wang, M.G. and Tang, L.X. "Genetic algorithm for the discrete time/cost trade-off problem in project network", *J. Northeastern Univ. (China)*, **21**(3), pp. 257-269 (2000).
8. Talbot, F.B. "Resource-constrained project scheduling with time-resource tradeoffs: The nonpreemptive case", *Management Science,* **28**, pp. 1197-1210 (1982).
9. Demeulemeester, E., De Reyck, B. and Herroelen, W. "The discrete time/resource trade-off problem in project networks: A branch-and-bound approach", *IIE Transactions,* **32**, pp. 1059-1069 (2000).
10. Vanhoucke, M., Demeulemeester, E. and Herroelen, W. "Discrete time/cost trade-offs in project scheduling with time-switch constraints", *Journal of the Operational Research Society*, **53**, pp. 741-751 (2002).
11. Babu, A.J.G. and Suresh, N. "Project management with time-cost and quality considerations", *European Journal of Operational Research,* **88**, pp. 320-327 (1996).
12. Khang, D.B. and Myint, Y.M. "Time, cost, quality trade-off in project management: A case study", *International Journal of project Management,* **17**(4), pp. 249-256 (1999).
13. El-Rayes, K. and Kandil, A. "Time-cost-quality trade-off analysis for highway construction", *Journal of Construction, Engineering Management*, **131**(4), pp. 477-485 (2005).
14. Tareghian, H.R. and Taheri, S.H. "On discrete time, cost and quality trade-off problem", *Applied Mathematics and Computation*, **181**, pp. 1305-1312 (2006).
15. Iranmanesh, H., Skandari, M.R. and Allahverdiloo, M. "Finding Pareto optimal front for the multi-mode time, cost, quality trade-off in project scheduling", *International Journal of Computer and Information and System Science and Engineering*, **2**, pp. 118-122 (2008).

16. Ravi Shankar, N., Raju, M.M.K. and Himabindu, P. "Discrete time, cost and quality trade off problem with renewable and non renewable resources", *International Journal of Computational Science and Mathematics*, **2**(1), pp. 285-290 (2010).

17. Mehdizade, E., and Mohsenian, O., " Solving time cost and quality trade off project problem using multi objective stochastic programming", Sharif Industrial Engineering and Managment Journal, **28-1** (2), pp. 103-11, (in Persain) (2013).

18. Zhang, H. and Xing, F. "Fuzzy-multi-objective particle swarm optimization for time-cost-quality tradeoff in construction", *Automation in Construction*, **19**, pp. 1067-1075 (2010).

19. Shahsavari Pour, N., Modarres, M. and Tavakkoli-Moghaddam, R. "Time-cost-quality trade-off in project scheduling with linguistic variables", *Word Applied Sciences Journal*, **18**(3), pp. 404-413 (2012).

20. Eskandari, H. and Geiger, C.D. "A fast Pareto genetic algorithm approach for solving expensive multiobjective optimization problems", *J. Heuristics*, **14**, pp. 203-241 (2008).

21. Deb, K., Pratap, A., Agarval, S. and Meyarivan, T.A. "Fast and elitist multiobjective genetic algorithm: NSGAII", *IEEE Trans*, **6**, pp. 182-197 (2002).

22. Kolisch, R., Sprecher, A. and Drexl, A. "Characterization and generation of a general class of resource-constrained project scheduling problems", *Instituten fur Betriebswirtschaftslehre der Universitat Kiel, Management Science*, **41**(10), pp. 1693-1703 (1992).

23. Zitzler, E., Deb, K. and Thiele, L. "Comparison of multiobjective evolutionary algorithms: Emprical results", *Evolutionary Computation Journal*, **8**(2), pp. 125-148 (2000).

24. Zitzler, E. "Eveloutionary algorithms for multi-objective optimization: Method and applications", P.h.D Thesis, Dissertation ETH NO. 13398, Swaziland Federal Institute of Technology Zorikh, Switzerland (1999).

25. Zitzler, E. and Thiele, L. "Multiobjective optimization using evolutionary algorithms a comparative case study", In A.E. Eiben, T. Back, M. Schoenauer and H. P. Schwefel (Eds.), *Fifth International Conference on Parallel Problem Solving from Nature (PPSN-V)*, Berlin, Germany, pp. 292-301 (1998).

26. Zitzler, E. and Thiele, L., *An Evolutionary Algorithm for Multiobjective Optimization: The Strength Pareto Approach*, Technical Report, No. 43, Zurich, Switzerland: Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) (1998).