نشریه علمی- پژوهشی علوم و فنون نقشه برداری، دوره چهارم، شماره ۱، مرداد ماه ۱۳۹۳

# یک راهکار ابتکاری تکاملی برای برنامه ریزی عامل های متحرک با رویکرد توازن به جای بهینه سازی چند هدفه

**محمد حسن وحیدنیا*[۱]، علی اصغر آل شیخ[۲]**

[۱] دانشجوی دکتری سیستم‌های اطلاعات مکانی- دانشکده مهندسی نقشه‌برداری- دانشگاه صنعتی خواجه نصیرالدین طوسی
vahidnia84@gmail.com

[۲] استاد گروه مهندسی نقشه‌بردرای- دانشکده مهندسی نقشه‌برداری- دانشگاه صنعتی خواجه نصیرالدین طوسی
(عضو قطب علمی مهندسی فناوری اطلاعات مکانی)
alesheikh@kntu.ac.ir

(تاریخ دریافت آبان ۱۳۹۲، تاریخ تصویب اردیبهشت ۱۳۹۳)

## چکیده

ساختارهای مبتنی بر عامل محیط مناسبی برای آزمودن فرضیه های پیچیده، مدلسازی و شبیه سازی طرح های پویای اطلاعاتی می باشند. در حیطه علوم مکانی نیز گرایش به استفاده از عامل های خودمختار و معقول، به دلیل واقع گرایی بیشتر در مسائل، افزایش یافته است. در این تحقیق به برنامه ریزی و توزیع وظیفه میان موجودیت های متحرک در فضای جغرافیایی با رویکرد عامل-محور پرداخته می شود. در ابتدا نشان داده می شود که عامل ها به طور منطقی مایل به رسیدن به بیشترین رضایت/سودمندی می باشند. برای دستیابی به رضایت گروهی در یک فعالیت جمعی، رسیدن به مفهوم توازن، برآمده از نظریه بازی ها، منطقی تر و واقع گرایانه تر از بهینه سازی چندهدفه به نظر می رسد. استدلال پیرامون این موضوع با بحث در مورد وابستگی و تزاحم سودمندی عامل های متحرک بیان می گردد. سپس با توجه به پیچیدگی محاسباتی بسیار بالا و نمایی تعیین توازن، یک الگوریتم ابتکاری غیرقطعی و کارآمد پیشنهاد می گردد. راهکار ارائه شده که همسو با روش های تکاملی می باشد، با چندین مجموعه داده ارزیابی گردید که نتایج حاکی از همگرایی مناسب، سرعت بالای الگوریتم، دقت مناسب تعیین توزان در مقایسه با روش قطعی (۸۰٪ در بدترین حالت) و دقت بالا در تعیین توازن با بیشترین بهینگی (۹۲٪ در بدترین حالت) می باشد.

**واژگان کلیدی :** سیستم اطلاعات مکانی، عامل های متحرک، نظریه بازی، توزان نش، الگوریتم تکاملی ابتکاری

* نویسنده رابط

## ۱- مقدمه

در حیطه علوم اطلاعات مکانی، ترکیب سیستم های اطلاعات مکانی[۱] و شبیه سازی عامل مبنا[۲] کاربرد قابل توجهی یافته اند [۱۷، ۱۸، ۱۹]. چنین حوزه ای از تحقیقات در بدو امر از میانه دهه نود میلادی آغاز گردیده است که طی آن اپستین و اکستل در سال ۱۹۹۶ مفهوم مدلسازی رفتار مردم در چگونگی رشد شهرها را به شکل مصنوعی مورد تحقیق قرار دادند [۷]. بطور کلی ترکیب سیستم اطلاعات مکانی و روش های مبتنی بر عامل، فرضیه سازی در مورد نحوه عملکرد یک سیستم پیچیده را فراهم می سازد [۵]. وجود قابلیت های لازم برای نمایش، مدیریت، بکارگیری همزمان توصیف و هندسه در این امر موثرند [۶].

از جمله کاربردهای مهم در رویکردهای مبتنی بر عامل، برنامه ریزی برای تعیین عملکرد عامل ها در فضای جغرافیایی و پیش بینی بازخورد آن در محیط می‌باشد. به عنوان نمونه در حیطه مدیریت بحران، امداد رسانی به مصدومین و مدل‌های پاسخگویی سریع، تحقیقات متعددی بر مبنای سیستم اطلاعات مکانی و مدلسازی عامل مبنا صورت گرفته است [۱۲، ۲۲]. تحقیقات دیگری نیز وجود دارند که تاکید بیشتری بر همکاری و مشارکت عامل‌ها در مسائل وظیفه محور مانند امداد رسانی، نجات و استراتژی-های نظامی داشته‌اند[۸، ۹، ۲۳].

رویکرد غالب در بسیاری از تحقیقات در حیطه برنامه ریزی و همکاری عامل ها، استفاده از مفاهیم تصمیم گیری مرکزی و مبتنی بر روش های بهینه سازی تک *هدفه* و چند *هدفه* می باشد. به طور مثال گراندینتی و همکاران در سال ۲۰۱۲ میلادی یک روش چند هدفه برای بهینه سازی مسئله یافتن بهترین استراتژی خدماترسانی به مشترکان توسط عامل های نقلیه توسعه دادند [۱۱]. قسیری و قنادپور در سال ۲۰۱۰ از روش های برنامه ریزی هدف[۳] و الگوریتم ژنتیک برای حل مسئله مسیریابی وسائط نقلیه در فضای چندهدفه استفاده نمودند [۱۰]. الگوریتم بهینه سازی کلونی مورچگان نیز پیشتر توسط بل و مک مولن در سال ۲۰۰۴ بکار گرفته شده بود [۴]. به طور مشابه ماریناکیس و ماریناکی در سال ۲۰۱۰ یک روش ترکیبی با استفاده از الگوریتم ژنتیک و ازدحام ذرات[۴] برای این مسئله استفاده کرده و به نتایج بهتری نسبت به الگوریتم ژنتیک دست یافتند [۱۳]. در برنامه ریزی لوجستیکی هلکوپترهای امداد رسانی، باربارسقلو و همکاران در سال ۲۰۰۲ از تحلیل های سلسله مراتبی و تصمیم گیری چندمعیاره برای حل مسئله به صورت چند هدفه استفاده نمودند [۲]. *شیو* در سال ۲۰۰۷ نیز از یک روش حل مسئله چندهدفه فازی در مسائل نمونه امداد رسانی فوری استفاده نمود و تأثیر عدم قطعیت را سنجید [۲۰].

در بسیاری از مسائل این چنینی که برنامه ریزی عامل‌ها در فضای جغرافیایی به شکل بهینه مطرح می باشند، به دلیل مدلسازی ساده تر، سودمندی حاصل از دست یابی به هدف یا رضایت برای یک عامل، مستقل از سودمندی عامل دیگر تعریف می شود. در حقیقت نکته مشترک در این تحقیقات و سایر تحقیقات مشابه عدم توجه به مفهوم سودمندی در فضای چند عامله و تزاحم یا تأثیر اتخاذ یک استراتژی توسط یک عامل بر میزان سودمندی یا دستیابی به هدف برای عامل دیگر می باشد [۱۵]. مدلسازی های دقیق تر ایجاب می کند که این تأثیر را نیز در نظر گرفت. به طور مثال فرض کنید در جریان امدادرسانی، یک عامل نقلیه باربری، یک طرح کاری را با توجه به مسیریابی بهینه و موقعیت های جمع آوری بارها در مسیرها و ایستگاه های ازپیش تعیین شده، برای خود تعریف نموده است. اگر به سودمندی این طرح بدون توجه به تأثیر و وابستگی به سایر عامل ها توجه کنیم، عامل همواره یک سودمندی ثابت باتوجه به طول مسیر، میزان بار، مصرف سوخت و سایر عوامل تأثیرگذار خواهد داشت. اما فرض کنیم که میزان سودمندی تابعی از عملکرد سایرین نیز باشد. مثلاً متقاطع یا هم مسیر بودن با یک عامل کمکی دوم موجب پیشبرد بهتر هدف و سودمندی بیشتر گردد یا به عکس نزدیکی با یک گروه عامل کمک های اولیه موجب تداخل، صرف زمان بیشتر و نهایتاً سودمندی کمتر برای عامل نقلیه بار به ازای همان طرح انتخابی گردد. اینجاست که دیگر روش های بهینه سازی تک هدفه و چند هدفه چندان کارآمد به نظر نمی رسند. چنین مفهومی گشاینده حیطه ای جذاب در تصمیم گیری عامل ها در غالب تئوری بازی می باشد که در سال های

---

۱ Geospatial Information System
۲ Agent-Based Modeling
۳ Goal Programming
٤ Particle Swarm

اخیر مورد توجه قرار رفته است. در حیطه علوم اطلاعات مکانی نیز در کاربردهایی همچون توسعه شهری و طراحی کاربری [۳] و طراحی سیستم های حمل و نقل شهری چند ساختی [۱۴] بیش از سایر کاربردها از ویژگی های عامل و مفاهیم رقابت، همکاری و وابستگی عامل ها استفاده نموده اند و به مدلسازی های واقع بینانه تری از محیط جغرافیایی دست یافته اند.

آنچه مد نظر این تحقیق می باشد استفاده از مفهوم توازن نش و تعریف ساختار بازی به جای بهینه سازی چندهدفه برای عامل های مکانی و با هدف برنامه ریزی می باشد. این مفهوم نشئت گرفته از نظریه بازی[۱] بوده و بیشتر در حیطه عامل های معاملاتی و رقابتی استفاده شده است. علت آن است که بنای این مفهوم بر کسب سودمندی متوازن و رضایت نسبی طرفین می باشد. در ادامه ابتدا راهکار پیشنهادی با ارائه ادله کافی برای اجتناب از بهینه سازی چندهدفه در مورد عامل های مکانی مطرح می گردد. به منظور جلوگیری از پیچیدگی محاسباتی بالای تعیین توازن در زمان وجود استراتژی و عامل های زیاد، یک راهکار محاسباتی غیرقطعی ابتکاری، همخوان با ویژگی های تکاملی، برای این منظور معرفی و بررسی می گردد. در انتها نتایج بدست آمده و مباحث تکمیلی ارائه می گردد.

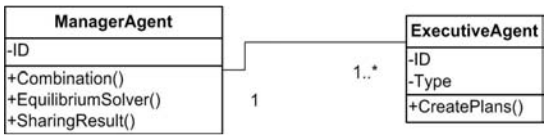
## ۲- راهکار پیشنهادی

### ۲-۱- طرح کلی مسئله

شکل انتزاعی مسئله بدین گونه است که تعدادی عامل نیمه خودمختار در فضای جغرافیایی موجود هستند که در منطقه حرکت نموده و وظایف محول شده را انجام می دهند تا هدف کلی سیستم چندعامله محقق شود. علت استفاده از عبارت نیمه خودمختار بدین دلیل است که اصولاً یک عامل کاملاً خودمختار ممکن است در هیچ فعالیت مشارکتی و وظیفه محور شرکت نکند. و اما نیمه خودمختار بودن یعنی یک عامل بر اساس مطلوبات خود و حوزه اختیاراتش چند طرح عملکرد دارد. هر طرح عملکرد برای یک عامل با توجه به اینکه سایر عامل ها چه طرح عملکردی انتخاب می نمایند، سودمندی یا نتیجه متفاوتی برای آن عامل حاصل می نماید. هدف تعیین یک استراتژی برای سیستم است به گونه ای که تمامی عامل ها از طرح عملکرد خود رضایت داشته باشند. فضای راه حل ها یک فضای گسسته در نظر رفته می شود که به طور معمول در برنامه ریزی های مکانی عامل های متحرک قابل استفاده است.

### ۲-۲- انواع عامل ها و ارتباط میان آن ها

در این تحقیق فارغ از طبقه بندی عامل ها بر اساس توانایی، آنگونه که در شکل ۱ ملاحظه می گردد، دو دسته عامل در نظر گرفته می شود: عامل هدایت کننده و عامل اجرایی. وابستگی میان این دو دسته به صورت متقابل و یک به چند می باشد که با توجه به یک هدف کلی تعریف می گردد. سیستم چندعامله سه مرحله را طی روند حل مسئله از طریق عامل هدایت کننده انجام می دهد. این سه مرحله عبارتند از ترکیب، حل مسئله و نهایتا به اشتراک گذاری با عامل های اجرایی. هر عامل اجرایی طرح های مطلوب خود را به عامل هدایت شونده انتقال می دهد و سپس در قسمت ترکیب، استراتژی های ممکن بر اساس تمام طرح های عملکردی عامل ها تشکیل می شود. در قسمت حل مسئله، استراتژی قابل قبول بر مبنای ایده توازن انتخاب شده و سپس هر عامل اجرایی موظف به اجرای طرح عملکرد برگزیده مربوط به خود می‌باشد.




شکل۱- رابطه، ویژگی ها و توابع عامل ها

### ۲-۳- ساختار وظیفه، طرح و برآیند

در این تحقیق یک ساختار پایه برای عامل های متحرک در کاربردهای وظیفه محور در نظر گرفته می شود. یک ساختار موقعیت $s$ در یک فضای مکانی با مختصات افقی و عمودی $x$ و $y$ در نظر می گیریم؛ یعنی $s:\langle x,y\rangle$ بطوریکه $x,y\in R$. یک *وظیفه* $t$ ساختاری است متشکل از شناسه وظیفه $i$ و برآیند یا *سودمندی* $o$ آن برای عامل و یا $t:\langle i,o\rangle$ به گونه ای که $o\in R$. ساختار کنش $a$ با دوگانه موقعیت $s$ و وظیفه $t$ یا $a:\langle s,t\rangle$ نشان داده می شود بدین معنی که عامل به سوی $s$ حرکت نموده و

[۱] Game Theory

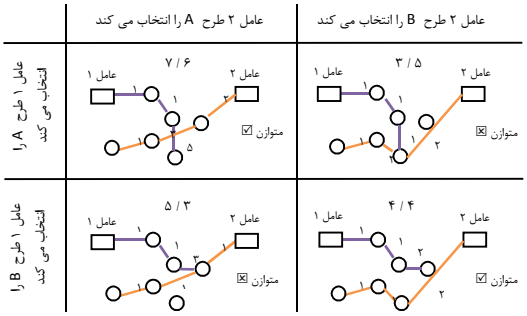
وظیفه $t$ را انجام می دهد. یک *طرح* $p$ مجموعه ای از $m$ کنش به صورت ترتیبی می باشد $p : \{a_j \mid j = 1,2,...,m\}$. برای یک طرح، میزان سودمندی یا برآیند طرح $p$ را طبق رابطه زیر محاسبه می نماییم.

$$f(p) = \sum_{j=1}^{m} o(t(a_j)); \quad a_j \in p \qquad (۱)$$

لازم به توضیح است که میزان سودمندی در دنیای علوم مکانی بسته به کاربرد، بسیار متنوع است. بطور مثال می توان در یک مسئله میزان مصرف انرژی کمتر و حرکت کمتر عامل متحرک را معادل سودمندی بیشتر در نظر گرفت درحالیکه در کاربردی دیگر میزان کار، حرکت و انرژی بیشتر و به تبعیت از آن به ثمر رساندن اهداف بیشتری از سیستم، عامل را به سودمندی یا رضایت بیشتری برساند.

## ۲-۴- انتخاب راه حل توازن

در اینجا، برای حل مسئله تقسیم وظایف عامل ها، یک ساختار بازی در نظر می گیریم [۱۶]. طبق تعریف، اگر در یک بازی یک استراتژی انتخاب شود و هیچیک از عامل ها از تغییر این استراتژی، در حالیکه طرح عملکرد سایرین ثابت نگاه داشته شود، به سودمندی بیشتری دست نیابد، چنین استراتژی یک توازن نش[۱] را تشکیل می دهد. در این تحقیق به منظور ساده سازی، تعداد طرح عامل ها مساوی در نظر گرفته می شود. شکل ۲ مثالی از یک ساختار بازی با دو عامل متحرک و دو طرح عملکرد برای هر یک را نشان می‌دهد. بنابراین ساختار بازی مشابه یک ماتریس ۲×۲ می‌باشد. هر المان ماتریس نشان دهنده یک استراتژی می‌باشد. در هر المان، مقادیر سودمندی برای عامل اول در مولفه سمت چپ و برای عامل دوم مولفه سمت راست آمده است. سودمندی کل بر اساس مجموع میزان سودمندی در هر یک از قطعات مسیر (برآیند انجام وظیفه) می باشد. آنگونه که ملاحظه می شود سعی شده تا به صورت نمادین نشان داده شود که هرگونه وابستگی یا تاثیر متقابل در طرح ها می تواند میزان سودمندی را برای یک عامل خاص و در یک طرح خاص تغییر دهد. بر اساس تعریف ارائه شده لازم است عامل ها هر یک برای انجام طرح عملکردی A هماهنگ شوند تا استراتژی حاصل یک توازن نش باشد. اگر هر دو عامل، طرح B خودشان را نیز اجرا کنند، مجدداً یک راه حل توازن برقرار خواهد بود. زیرا اگرچه سودمندی هریک از مقدار بهینه ۶ و ۷ فاصله دارد اما هیچیک تمایل به تغییر استراتژی به دلیل کاهش سودمندی در استراتژی دیگر (از مقدار ۴ به ۳) ندارند.




شکل ۲- یک نمونه از ساختار بازی نشان دهنده سودمندی نسبی عامل های متحرک در ترکیب های دوتایی عامل دوم/ عامل اول. راه حل توازن نش زمانی است که هر دو عامل طرح A و یا هردو طرح B را انتخاب نمایند. برآیند کنش های تشکیل دهنده یک طرح روی آن ذکر شده است.

برای تبیین رسمی، یک بازی را به صورت $(S, f)$ نشان می دهیم و فرض می کنیم که $n$ عامل در این بازی شرکت دارند. اگر $S_i$ مجموعه استراتژی ها برای عامل $i$ باشد، $S=S_1\times S_2...\times S_n$ مجموعه پروفایل استراتژی ها برای تمامی عامل ها می باشد و $f=(f_1(x), ..., f_n(x))$ یک تابع سودمندی است به گونه ای که $x \in S$. همچنین $x_i$ نشان دهنده یک پروفایل استراتژی برای عامل $i$ می باشد و $x_{-i}$ یک پروفایل استراتژی برای تمامی عامل ها به غیر از عامل $i$ می باشد. زمانی که هر عامل $i \in \{1, ..., n\}$ یک طرح استراتژیک $x_i$ را انتخاب می کند، نتیجه آن یک پروفایل استراتژی همانند $x=(x1, ..., xn)$ می شود و عامل $i$ به سودمندی $f_i(x)$ دست می یابد. میزان سودمندی به پروفایل استراتژی انتخاب شده بستگی دارد. یک پروفایل استراتژی $x^* \in S$ توازن نش است، یا $NE(x^*)$، اگر و تنها اگر هیچ انحراف یکسویه در آن استراتژی توسط یک عامل، برای آن عامل سودمند نباشد یا به شکل ریاضی داشته باشیم [۲۱]:

$$NE(x^*) \text{ iff } \forall i, x_i, x_i \neq x_i^*: \quad f_i(x_i^*, x_{-i}^*) \geq f_i(x_i, x_{-i}^*) \qquad (۲)$$

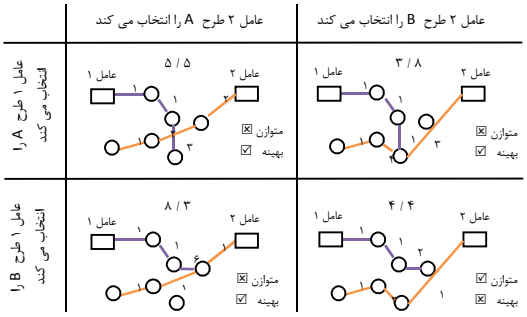
۱ Nash Equilibrium

## ۲- ۵- ترجیح منطقی توازن بر بهینگی چند هدفه

بـایـد توجه داشـــت که توازن به معنی بهینگی پارتو[۱] نیسـت [۱۶]. یک استراتژی بهینه است در صورتیکه هیچ اســتراتژی غالبی نسـبت به آن وجود نداشــته باشــد و به عبارت بهتر $x^* \in S$یک استراتژی است که به صورت چند هدفه (برای تمامی عامل ها) بهینه است، یا $\mathcal{PO}(x^*)$، اگر و تنها اگر شرایط زیر برقرار باشند:

$$\mathcal{PO}(x^*) \text{ iff } \nexists\, x, x \neq x^*. \forall\, i: f_i(x) \geq f_i(x^*) \qquad (۳)$$

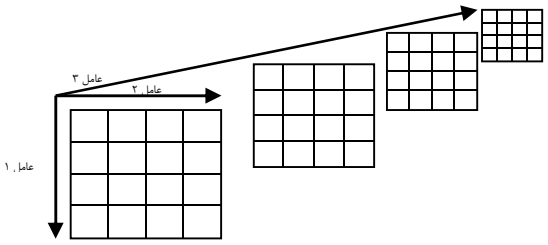
با توجه به این تعریف بسیار محتمل است که یک راه حل بـه طور چنـدهـدفـه بهینه باشـــد، اما عامل ها از انتخاب آن اســـتراتژی ناراضـــی بوده و متمایل به تغییر اســـتراتژی برای رسـیدن به سـودمندی بیشـتر باشند. بطور مثال فرض کنیم اســـتراتژی عـامل های متحرک به ســـاختار بازی شـــکل ۳ بیانجامد. در این ماتریس تمامی اسـتراتژی ها به غیر از حالتی کـه هر دو عـامـل طرح B را انتخاب می کنند بهینه پارتو می باشـند. هر چند انتخاب طرح B توسـط هر دو عامل منجر به توازن می گردد، امـا این راه حل نســـبت به حالتی که هر دو عامل A را انتخاب می کنند، مغلوب اســـت و بنابراین یک راه حل بهینه نیسـت. با انتخاب طرح A توسط دو عامل، بهینگی وجود دارد اما توازن برقرار نیســـت. اما حالتی را متصـــور می شـــویم که یکی از عامل ها طرح A و دیگری طرح B خود را انتخاب می کند. اســـتراتژی حاصـــل مجدداً راه حل مســـئله بهینگی چند هدفه می باشـد زیرا هیچ استراتژی غالبی نسبت بـه آن وجود ندارد. اما توازنی در این دو راه حل وجود ندارد و عامل ها تمایل دارند اســـتراتژی خود را تغییر دهند. در واقع با وجود بهینه بودن راه حل، یکی از عامل ها ســـودمندی زیاد و دیگری سـودمندی کمی بدسـت می آورند که عادلانه نیست (در حالیکه با تغییر طرح خود می تواند سـودمندی را افزایش دهـد). بـه این دلیل اســـت که در این تحقیق با نگرش موثر بودن تصـــمیم‌گیری یک عامل متحرک بر میزان ســـودمندی عـامـل هـای دیگر، روش تصـــمیم گیری چند هدفه موجب رضـایت گروه عامل ها نخواهد شد. در نتیجه توازن نش مورد استفاده قرار خواهد گرفت.




شکل ۳- تفاوت بهینگی چندهدفه و توازن در یک نمونه از ساختار بازی. راه حل توازن نش زمانی است که هر دو عامل طرح B را انتخاب نمایند. تمامی استراتژی ها به غیر از حالتی که هر دو عامل طرح B را انتخاب می کنند بهینه پارتو می باشند.

## ۲-۶- یک فرآیند غیر قطعی ابتکاری

تمامی شیوه های مطرح شده برای تعیین قطعی توازن خالص نش در نیم قرن اخیر ماهیت ترکیبیاتی[۲] دارند و متاسفانه هیچ یک، به ویژه برای بازی های با ابعاد بالا، بهینه به نظر نمی رسند [۱۵]. در یک بازی با $n$ عامل و $\rho$ طرح به ازای هر عامل، تعداد اعضای ماتریس چند بعدی بازی از مرتبه نمایی $\rho^n$ خواهد شد (شکل ۴). هر عضو ماتریس یک ساختار $n$-تایی دارد. با توجه به مثال هایی که در قسمت های قبلی مطرح گردید، در یک روش خالص، تعداد مقایسات در یک درایه ماتریس، برای آنکه متوجه شویم توزان نش در آن برقرار است، $n.(\rho\text{-}1)$ خواهد بود. در نتیجه مرتبه محاسبات برای کل ماتریس به صورت $O(n.\rho^{n+1})$ برآورد می شود.




شکل ۴- نمایش یک ماتریس چندبعدی بازی برای ۳ عامل و ۴ طرح به ازای هر عامل.

در این تحقیق یک روش غیر قطعی ابتکاری معرفی و ارزیابی می گردد تا سرعت یافتن توازن نسبی تا حد امکان

۱ Pareto optimality

۲ Combinatorial

افزایش یابد. در راهکار پیشنهادی از ایده محاسبات تکاملی[۱] در هوش مصنوعی[۲] استفاده می کنیم. از ویژگی های اصلی یک روش تکاملی اولاً تعیین مجموعه ای از راه حل های اولیه می باشد؛ ثانیاً از عملگرهایی برای ایجاد راه حل های بعدی یا نسل های بعدی استفاده می نماید؛ ثالثاً از یک تابع برازش برای تعیین کیفیت راه حل ها بهره می برد؛ رابعاً از فرآیندهای تکراری استفاده می نماید [۱]. باید توجه داشت که مسئله مطرح شده در این تحقیق آنگونه که توضیح داده شد، با بهینه سازی تک هدفه یا چند هدفه قابل حل نیست و بنابراین الگوریتم های تکاملی توانمندی چون ژنتیک، NSGA و موارد مشابه قابل استفاده نیستند. برای راه حل‌های اولیه از مفهوم استراتژی غالب[۳] و انتخاب تصادفی بهره می گیریم. در قسمت ایجاد راه حل های نسل بعدی سه عملگر به نام های همسایگی، جهش و تقاطع استفاده می شوند. مبنای تابع برازش، کمینه کردن اختلاف میان تعداد کلی عامل ها است با تعداد عامل هایی که در شرایط برقراری توازن قرار دارند. درنهایت شرط توقف، رسیدن به جواب، یا پایان تکرارها و یا زمان مد نظر می باشد.

### ۲-۶-۱- ایجاد راه حل اولیه

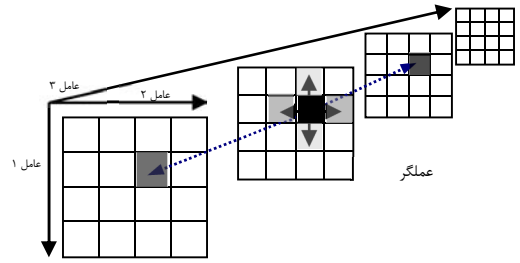
یک استراتژی برای یک عامل غالب است اگر بدون توجه به انتخاب سایر عامل ها همواره برای آن عامل در اولویت قرار داشته و سودمندی بیشتری حاصل نماید [۲۱]. اگر همه عامل ها یک استراتژی غالب داشته باشند، آنگاه طبیعی است که هریک استراتژی غالب را انتخاب نمایند و در این حالت ما به یک توازن به نام توازن استراتژی غالب می رسیم. بنابراین، یک حدس اولیه در راهکار پیشنهادی جهت یافتن توازن آن است که به دنبال استراتژی های غالب یا نزدیک به غالب بگردیم و یا دست کم از استراتژی‌های مغلوب صرف نظر نماییم. هرچند باید توجه داشت که یافتن استراتژی غالب در صورت وجود، مجددا الگوریتم کارآمدی ندارد و می بایست راهکاری ابتکاری برای نزدیک شدن به آن یافت. در فرآیند انتخاب جواب های اولیه انتخاب های تصادفی[۴] نقش دارند اما علاوه برآن، بیشینه بودن سودمندی یک استراتژی خاص برای یک عامل شانس نزدیکی به استراتژی غالب یا نسبتاً غالب (برقراری غالب بودن برای بیشتر عامل ها) را افزایش می دهد. برای تولید یک راه حل اولیه، ابتدا عامل اول به صورت تصادفی انتخاب می شود. این عامل را i می نامیم. در مرحله بعد، طرح انتخابی عامل ها، به غیر از i، بطور تصادفی از میان طرح های ممکن مشخص می گردد که آن را با $x^r_{-i}$ نشان می دهیم. حال طرح سایر عامل ها غیر از $i$ را ثابت فرض نموده و سپس طرح مربوط به عامل $i$ را با پیدا نمودن بیشترین سودمندی در میان طرح هایش یا $\mathrm{Max}\{f_i(x_i, x^r_{-i})\}$ پیدا می نماییم و آن را $x_i^*$ می نامیم. به طور مشابه، عامل دوم را به صورت تصادفی انتخاب می کنیم و آن را $j$ می نامیم به طوریکه $i\neq j$. سپس طرح مربوط به سایر عامل‌ها، به غیر از عامل های طرح دار، یعنی عامل $i$ (که در مرحله قبل دارای طرح شد) و عامل کنونی $j$ ، به صورت تصادفی انتخاب می شوند. طرح عامل $j$ نیز بر اساس بیشترین سودمندی $\mathrm{Max}\{f_j(x_i^*, x_j, x^r_{-i,j})\}$ تعیین کرده و $x_j^*$ نامگذاری می کنیم. به طور مشابه این عمل تا پایان کل مجموعه عامل ها ادامه می یابد تا نهایتاً یک استراتژی اولیه به صورت $S^*=x_1^* \times x_2^* \times \ldots \times x_n^*$ حاصل شود. هر راه حل اولیه بدین شکل بدست می آید و بنابراین می توان مجموعه ای از راه حل های اولیه بدین طریق ایجاد نمود. مجموعه راه حل های اولیه را با $\kappa$ نشان می دهیم.

### ۲-۶-۲- اعمال عملگرها

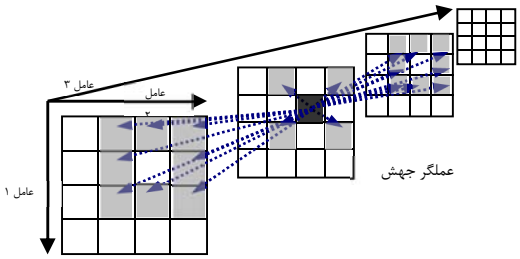
برروی هریک از راه حل ها در $\kappa$ و طی هر مرحله از فرآیند تکرار، یکی از دو عملگر همسایگی یا جهش اعمال می گردند. همچنانکه پیش از آن بر روی بخش کوچکی از راه حل ها (ζ) عملگر تقاطع اعمال می گردد. یک درایه همسایه نسبت به درایه کنونی در ماتریس بازی، از مجموعه درایه های اطراف انتخاب می شوند که در آن، تنها طرح یکی از عامل ها با طرح کنونی اش متفاوت باشد (شکل ۵). بنابراین هر درایه یا عضو ماتریس حداکثر $2n$ همسایه دارد. عملگر همسایگی، راه حل فعلی را به یک راه حل همسایگی، که دست کم سودمندی بیشتری برای یک عامل دارد، انتقال می‌دهد. در شکل، راه حل های همسایه با رنگ خاکستری کمرنگ، متوسط و تیره به ترتیب برای عامل اول، دوم و سوم نشان داده شده اند. این بدین معنی است که عملگر همسایگی همواره به دنبال رسیدن به حالتی است که یک عامل به سودمندی بیشتری دست یابد و مایل نباشد آن استراتژی را تغییر دهد. وقتی عامل های

۱ Evolutionary computation
۲ Artificial Intelligence
۳ Dominant strategy
٤ Randomness

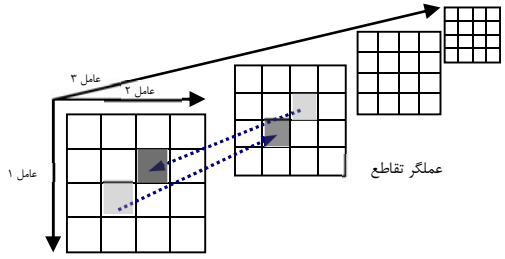
بیشتری این حالت را داشته باشند به توازن نزدیک شده‌ایم. زمانی که چنین همسایگی موجود نباشد، یک عملگر جهش فضای جستجو را تغییر می‌دهد. شعاع تغییر طرح در عملگر جهش بین یک واحد مثبت تا منفی برای تمامی عامل‌ها در ماتریس بازی، به غیر از عناصر همسایگی، می باشد (شکل ۶). در اعمال تقاطع، دو استراتژی اولیه انتخاب گردیده و بطور تصادفی طرح یکی از عامل ها بین دو استراتژی جابجا می‌گردد. به طور مثال، در شکل دو استراتژی با رنگ‌های روشن بر اساس جابجایی در طرح عامل سوم به دو استراتژی با رنگ تیره تبدیل شده‌اند (شکل ۷).




شکل ۵- نمایش همسایگی برای یک عنصر، در یک ماتریس چندبعدی بازی برای ۳ عامل و ۴ طرح به ازای هر عامل. جهت نماهای داخلی برای نشان دادن همسایگی یک درایه بکاربرده شده اند. عملگر همسایگی راه حل فعلی را به یک راه حل همسایگی که دست کم سودمندی بیشتری برای یک عامل دارد انتقال می دهد.




شکل ۶- نمایش فضای جهش برای یک عنصر، در یک ماتریس چندبعدی بازی برای ۳ عامل و ۴ طرح به ازای هر عامل. جهت نماهای داخلی برای نشان دادن جهش یک درایه بکاربرده شده اند. عملگر جهش راه حل فعلی را به یک راه حل تصادفی غیر همسایگی انتقال می دهد.




شکل ۷- نمایش تقاطع برای یک عنصر، در یک ماتریس چندبعدی بازی برای ۳ عامل و ۴ طرح به ازای هر عامل. دو استراتژی با رنگ های روشن بر اساس جابجایی در طرح عامل سوم به دو استراتژی با رنگ تیره تبدیل شده اند.

### ۲-۶-۳- تابع برازش

همانگونه که عنوان شد، مبنای تابع برازش کمینه کردن اختلاف میان تعداد کلی عامل‌ها است با تعداد عامل‌هایی که در شرایط برقراری توازن قرار دارند. در حقیقت الگوریتم همواره راه حل ها را مورد آزمایش قرار می دهد و به ازای کمترین مقدار تابع برازش، آن راه حل را به عنوان بهترین راه حل حفظ می کند. اگر راه حلی یافت شد که با بهترین راه حل کنونی از نظر مقدار توازن برابر بود، راه حل بهینه تر به عنوان بهترین جواب انتخاب می شود. مقادیر توازن در یک نسل از راه حل ها در مجموعه ای به نام $\mathcal{B}$ نگهداری می شود و بر اساس آن مقدار برازش محاسبه می شود.

### ۲-۶-۴- شرط توقف

الگوریتم فرآیندی تکراری را تا رسیدن به جواب نهایی طی می کند. بیشترین تعداد تکرارها را با $\gamma$ نشان می دهیم. در صورت عدم تفاوت فاحش ($\varepsilon$)، میان مقدار برازش در کل مجموعه راه حل های یک نسل با نسل بعدی، الگوریتم متوقف می شود. در غیر این صورت تعداد تکرارها یا محدودیت زمانی می تواند برای توقف اعمال شود. استراتژی یافت شده را با $\mathcal{S}^*$ نشان می دهیم. شکل کلی الگوریتم مطرح شده در این تحقیق را در الگوریتم ۱ مشاهده می نمایید.

**Non-deterministicHeuristicNE** (Game Matrix: $\mathcal{G}$) best solution: $\mathcal{S}^*$, best fitness: $\mathcal{B}^*$

```
Select a small set of initial solutions from G based on dominance idea and insert into κ
Loop for γ times {
  Apply Crossover operator for ζ fraction of κ
  Loop for every S ∈ κ {
    Evaluate Nash equilibrium or NE of S and keep it in the set B
    If S has a neighborhood with higher payoff for one agent {
      Apply Neighborhood operator on S }
    Else {
      Apply Mutation operator on S
    }
    Keep track of best solution in S* and best fitness denoted by B*
  }
  If the total difference of current B and the previous one is less than the ε {
    Exit loop
  }
}
```

الگوریتم ۱- نمایش الگوریتم پیشنهادی بر اساس شبه کدها

## ۳- ارزیابی و مباحث تکمیلی

### ۳-۱- آماده سازی داده ها

ارزیابی راهکار ارائه شده، براساس داده های شبیه سازی شده عامل‌های متحرک انجام گردید. بر این اساس ۸ مجموعه داده و به عبارت بهتر ۸ بازی برای آزمایش الگوریتم آماده سازی گردید. در ۳ مجموعه اول ۵ عامل، ۳ مجموعه دوم ۶ عامل و ۲ مجموعه آخر ۷ عامل در نظر گرفته شد. برای هر عامل ۸۰ طرح در نظر گرفته شد و در بین بازی ها توزیع شد. در هر طرح ایستگاه‌هایی عملیاتی برای عامل ها در نظر گرفته شده و برای هرایستگاه میزانی از برآیند یا ارزش کاری در نظر گرفته شد. میزان سودمندی حرکتی با توجه به گراف وزن دار، شبکه و مسیریابی بهینه انجام گرفت. عدم تداخل مکانی طرح با طرح دیگر عامل‌ها، میزان وابستگی کاری عامل‌ها به یکدیگر از لحاظ مکانی و هم مسیری، که منجر به تأثیرات متقابل عامل‌ها می‌گردید نیز، در تعیین سودمندی بکارگرفته شد. نکته قابل توجه بالا بودن حجم داده‌ها می باشد. بطور مثال در کمترین حالت برای ۵ عامل و ۱۰ طرح نیاز به محاسبه ۵۰۰۰۰۰ مقدار مجزا به عنوان سودمندی و تشکیل ماتریس بازی بود. با توجه به حجم بالا و چندبعدی بودن داده‌ها امکان نمایش آن‌ها در مقاله میسر نمی باشد. شکل ۸ نمایی مکانی از داده‌های پایه استفاده شده برای بخشی از منطقه ۳ تهران می باشد. آماده سازی داده‌های پایه، مسیریابی بهینه و تأثیر مکانی طرح‌های حرکتی (مانند شعاع فاصله، تداخل مسیرها) در نرم افزار ArcGIS® و به کمک ابزار Network Analyst امکان‌پذیر بود. سایر محاسبات سودمندی و تشکیل ماتریس بازی و پیاده‌سازی الگوریتم در نرم افزار MATLAB® انجام پذیرفت.

شکل ۸- نمونه داده های پایه استفاده شده از شهر تهران؛ نمونه طرح حرکتی عامل ها و ایستگاههای عملیاتی

## ۳-۲- اجرای الگوریتم و بررسی همگرایی

به منظور تعیین کارآیی اجرای الگوریتم برای یافتن استراتژی مناسب عامل ها، هر یک از ۸ مجموعه داده به عنوان ورودی به الگوریتم معرفی گردید. مشخصات ورودی ها و تنظیمات مربوط به الگوریتم در جدول۱ مشاهده می گردد. تلاش گردید مجموعه هایی با ویژگی‌های متفاوت به عنوان ورودی در نظر گرفته شوند تا از میزان صحت نتایج اطمینان حاصل شود. در تمامی مجموعه های داده مقدار ε برابر ۰/۱ و نرخ تقاطع برابر ۰/۰۸ قرار داده شد. نتایج اجرای الگوریتم ها در نمودارهایی که نشان دهنده تعداد تکرار و مقدار برازش در مجموعه راه حل یک نسل می باشد، گردآوری شده است (شکل ۹). مطابق شکل، نتایج کاملاً قابل قبول و بهبود برازش تا زمان همگرا شدن راه حل ها به سمت توازن بیشتر، کاملاً مشخص است. الگوریتم عموماً خیلی زودتر از تعداد تکرار اولیه معرفی شده به نتیجه دلخواه رسیده است. لازم است توجه گردد که میزان برازش در مجموعه لزوماً مقداری نزدیک به صفر نیست و این بستگی به ماتریس بازی دارد. اگر یک بازی به تعداد راه حل های یک نسل، توزان داشته باشد آنگاه مقدار برازش کل باید به صفر برسد. چنین ویژگی به طور معمول در هیچ بازی اتفاق نمی افتد. همانگونه که در شرح الگوریتم توضیح داده شد، اگر در متوازن ترین راه حل های به دست آمده، تعداد عامل هایی که عدم تمایل به تغییر استراتژی را داشته باشند برابر باشد، آنگاه راه حلی که نسبت به سایر راه حل ها بهینه تر است به عنوان خروجی الگوریتم برگزیده می شود.

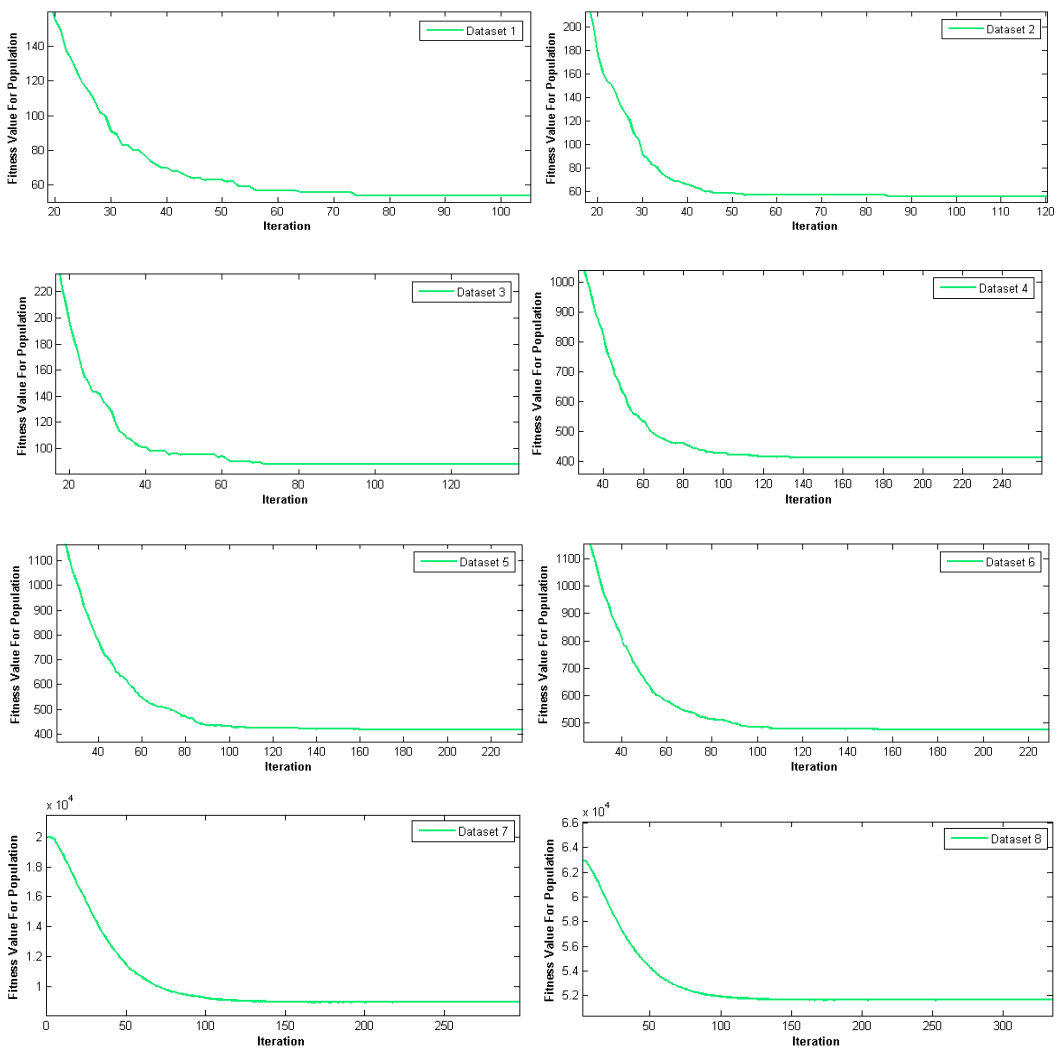
جدول ۱- ویژگی ورودی های الگوریتم و تنظیمات اولیه

| شناسه مجموعه داده یا بازی | تعداد عامل | تعداد طرح برای هر عامل | تعداد تکرارهای اولیه | تعداد جواب های اولیه |
|---|---|---|---|---|
| ۱ | ۵ | ۸ | ۱۲۰۰ | ۴۰۰ |
| ۲ | ۵ | ۱۰ | ۱۵۰۰ | ۵۰۰ |
| ۳ | ۵ | ۱۲ | ۱۸۰۰ | ۶۰۰ |
| ۴ | ۶ | ۸ | ۳۰۰۰ | ۱۴۰۰ |
| ۵ | ۶ | ۱۰ | ۳۵۰۰ | ۱۸۰۰ |
| ۶ | ۶ | ۱۲ | ۴۰۰۰ | ۲۰۰۰ |
| ۷ | ۷ | ۸ | ۱۴۰۰۰ | ۱۰۰۰۰ |
| ۸ | ۷ | ۱۲ | ۱۶۰۰۰ | ۱۲۰۰۰ |

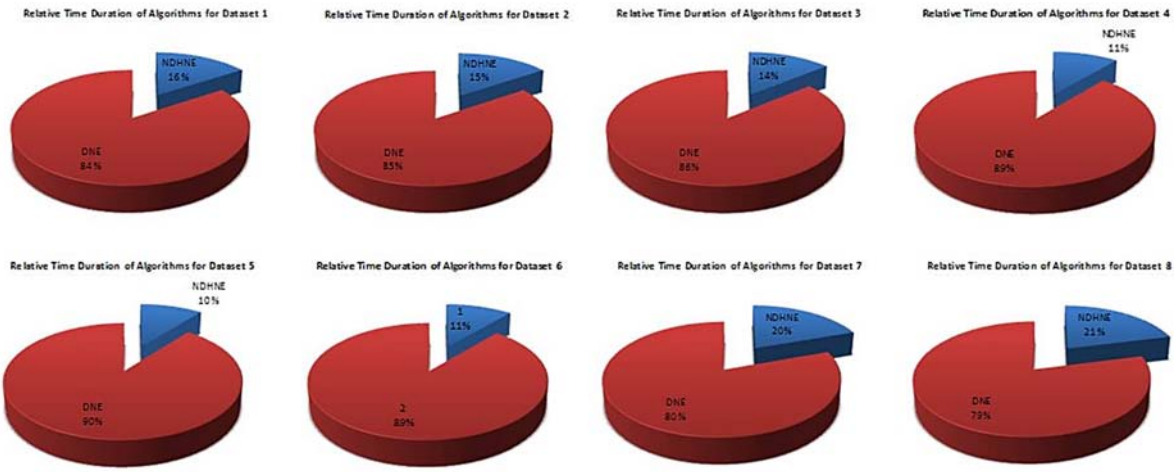
## ۳-۳- مقایسه با جواب های روش قطعی

علاوه بر راهکار پیشنهادی، الگوریتمی برای تعیین قطعی توازن توسعه یافت. همه ماتریس های بازی، علاوه بر الگوریتم ابتکاری در الگوریتم قطعی نیز مورد آزمایش قرار گرفتند. برمبنای این مقایسه، نتایج بدست آمده برای الگوریتم پیشنهادی، مطلوب و حتی فراتر از انتظار ارزیابی گردید. در مقایسات این قسمت، الگوریتم غیرقطعی ابتکاری تحقیق را با NDHNE و الگوریتم قطعی را با DNE نشان می دهیم و مجموعه داده ها یا بازی ها را بر اساس شناسه آن ها مقایسه می کنیم. شکل ۱۰ نتیجه مقایسه زمان اجرای الگوریتم ها را نشان می دهد. همانگونه که ملاحظه می شود، با اختلاف زمانی بسیار محسوسی NDHNE نسبت به DNE در تمام مجموعه های داده سریعتر عمل نموده است. در بدترین حالت این نسبت ۷۹ به ۲۱ حاصل شده است. زمان واقعی اجرای الگوریتم در مجموعه آخر حدوداً ۲ ساعت و ۴۰ دقیقه در الگوریتم قطعی DNE به طول انجامید در حالیکه برای الگوریتم ابتکاری NDHNE حدوداً در ۴۱ دقیقه انجام شد.

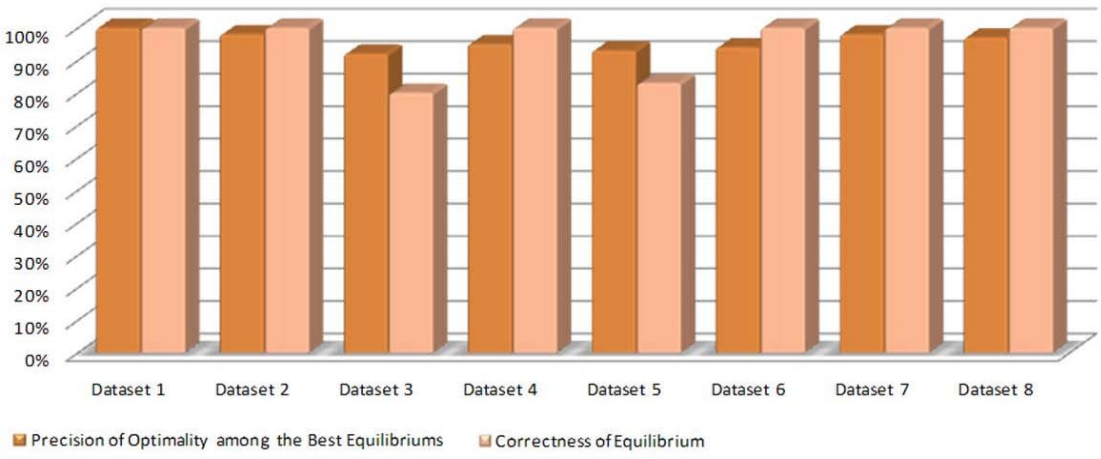
اما نکته مهم آن است که صرفه جویی در زمان به ازای کاهش نامطلوب دقت وصحت نتایج نباشد. بنابراین در این قسمت، در دو حالت دیگر DNE و NDHNE را مقایسه نمودیم. با توجه به اینکه DNE به طور قطعی عمل کرده و راه حل بازی با بیشترین توازن ممکن برای عامل ها و با بیشترین بهینگی در میان راه حل ها با توازن یکسان را به عنوان خروجی ارائه می دهد، در این قسمت تعیین می کنیم که روش پیشنهادی NDHNE تا چند درصد به این صحت و دقت نزدیک بوده است. نتایج در نمودار واحد شکل ۱۱ و به تفکیک هر ۸ مجموعه ارائه گردیده است. نمودارها نشان می دهند که دقت مناسبی در تعیین توزان با توجه به نتایج روش قطعی (۸۰٪ در بدترین حالت) و در تعیین توازن با بیشترین بهینگی (۹۲٪ در بدترین حالت) حاصل گردیده است. دقت ۸۰٪ در تعیین توازن در مجموعه داده سوم به این معنی است که طبق راه حل قطعی، یک استراتژی وجود دارد که در آن ۵ عامل در حالت توازن باشند اما راه حل غیر قطعی به جوابی بهتر از ۴ عامل نرسیده است.




شکل ۹- روند همگرایی و الگوریتم ها و برازش برای هر ۸ بازی




شکل ۱۰- مقایسه زمان نسبی اجرای الگوریتم برای روش قطعی DNE و روش ابتکاری تحقیق NDHNE به تفکیک مجموعه های داده.




شکل ۱۱- نمودار میزان صحت بهترین توازن و مقدار بهینگی راه حل توزان در NDHNE نسبت به DNE

## ۳-۴- پیچیدگی الگوریتم

به لحاظ محاسباتی، الگوریتم مطرح شده در قسمت ایجاد جواب های اولیه برای تمامی عامل ها اجرا شده و هر مرتبه بیشترین سودمندی ها را تعیین می نماید تا یک استراتژی را انتخاب نماید یا به عبارت بهتر $\forall i\ f_i(x_i, x^r_{-i})$ در نتیجه، در صورت استفاده از *رتبه بندی ادغامی*[۱] می توانیم یک مرتبه در حدود $O(n.\rho.\log\rho)$ برای این قسمت در نظر بگیریم. در نتیجه مرتبه کلی الگوریتم برای $k$ راه حل اولیه $O(k.n.\rho.\log\rho)$ خواهد شد. اما محاسبات فرآیند تکراری الگوریتم با توجه به بررسی توازن نش، که در یک مولفه ماتریس زمانی در مرتبه $O(n.\log\rho)$ به طول می انجامد، معادل $O(\gamma.k.n.\log\rho)$ خواهد شد. محاسبات عملگرها به دلیل تعداد ثابت، در ترم کلی بی تأثیر خواهند بود و نهایتاً مرتبه کلی الگوریتم $O(k.n.\rho.\log\rho + \gamma.k.n.\log\rho)$ برآورد می شود که عموماً به دلیل تکرارهای زیاد نسبت به تعداد طرح های انتخابی می توان از عبارت سمت چپ صرف نظر نمود. در صورت جوابگویی الگوریتم در تعداد تکرار مناسب و برای تعداد مناسب راه حل اولیه، می توان در زمان پلی نومیال به جواب رسید که بسیار امیدبخش تر از زمان نمایی $O(n.\rho^{n+1})$ در روش قطعی می باشد.

## ۴- نتیجه گیری

در این تحقیق نشان داده شد که رضایت یا سودمندی های وابسته در عامل های متحرک منجر به ناکارآمدی روش های بهینه سازی چندهدفه در برنامه ریزی می شود. در این شرایط منطق حکم می کند که عامل ها به دنبال توازن باشند به گونه ای که هیچیک مایل به تغییر استراتژی انتخاب شده نباشد. برای این منظور یک روش ابتکاری با رویکرد تکاملی پیشنهاد گردید که بتواند به شکلی کارآمد و با دقت مناسب جواب توازن را بیابد. نتایج حاکی از آن است که الگوریتم معرفی شده با قابلیت اطمینان بالایی قابل جایگزینی با راه حل قطعی است؛ به خصوص اگر زمان بسیار کوتاه تر رسیدن به یک جواب نسبتاً قابل اطمینان بر زمان طولانی دستیابی به یک جواب کاملاً صحیح ارجحیت داشته باشد. صحت نتایج نیز در مقایسه با نتایج روش قطعی قابل قبول ارزیابی گردید. ساختاردهی چگونگی تعیین سودمندی عامل های مکانی، حل مسائل مشابه در فضای پیوسته به جای گسسته، و بهبود هر چه بیشتر الگوریتم و راهکار پیشنهادی در دستور کارهای تحقیقاتی آینده قرار می گیرند.

## مراجع

[1] Ashlock, D., (2006). "Evolutionary Computation for Modeling and Optimization". Berlin: Springer-Verlag.

[2] Barbarosoğlu, G., Özdamar, L. and Çevik, A., (2002). "An interactive approach for hierarchical analysis of helicopter logistics in disaster relief operations". European Journal of Operational Research, 140 (1), 118–133.

[3] Behzadi, S., Alesheikh, A., (2013). "Introducing a novel model of belief–desire–intention agent for urban landuse planning". Engineering Applications of Artificial Intelligence, 26 (9), 2028–2044.

۱ Merge Sort

[4] Bell, J.E. and McMullen, P.R., (2004). "Ant colony optimization techniques for the vehicle routing problem". Advanced Engineering Informatics, 18 (1), 41–48.

[5] Benenson, I. and Torrens, P.M., (2004). "Geosimulation: object-based modeling of urban phenomena". Computers, Environment and Urban Systems, 28 (1/2), 1-8.

[6] Castle, C.J.E. and Crooks, A.T., (2006). "Principles and concepts of agent-based modelling for developing geospatial simulations". Working Paper 110, Centre for Advanced Spatial Analysis, University College London, UK. Available from: http://www.casa.ucl.ac.uk/working_papers/paper110.pdf.

[7] Epstein, J.M. and Axtell, R.L., (1996). "Growing artificial societies: Social science from the bottom up". Cambridge, MA: The MIT Press.

[8] Frank, A.U., (1996). "Qualitative spatial reasoning: cardinal directions as an example". International Journal of Geographic Information Systems, 10 (3), 269-290.

[9] Gagne´, D., Pang, W. and Trudel, A., (1997). "A spatio-temporal logic for 2D multi-agent problem domains". International Journal of Expert Systems with Applications, 12 (1), 141-145.

[10] Ghoseiri, K. and Ghannadpour, S.F., (2010). "Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm". Applied Soft Computing, 10 (4), 1096–1107.

[11] Grandinetti, L., Guerriero, F., Laganá, D. and Pisacane, O., (2012). "An optimization-based heuristic for the Multi-objective Undirected Capacitated Arc Routing Problem". Computers & Operations Research, 39 (10), 2300–2309.

[12] Guo, D., Ren, B. and Wang, C., (2008). "Integrated agent-based modeling with GIS for large scale emergency simulation". In: L. Kang, Z. Cai, X. Yan and Y. Liu, eds. Advances in Computation and Intelligence (ISICA 2008). Berlin: Springer-Verlag, 618-625.

[13] Marinakis, Y. and Marinaki, M., (2010). "A hybrid genetic – Particle Swarm Optimization Algorithm for the vehicle routing problem". Expert Systems with Applications, 37 (2), 1446–1455.

[14] Miller H. J., (2011). "Collaborative mobility: using geographic information science to cultivate cooperative transportation systems". Procedia - Social and Behavioral Sciences, 21, 24–28.

[15] Nisan, N., Roughgarden, T., Tardos, E. and Vazirani, V., (2007). "Algorithmic Game Theory". Cambridge University Press.

[16] Osborne, M., (2004). "An introduction to game theory". Oxford University.

[17] Parker, D.C., (2005). "Integration of geographic information systems and agent-based models of land use: Challenges and prospects". In: D.J. Maguire, M. Batty and M. Goodchild, eds. GIS, spatial analysis and modelling. Redlands: ESRI Press, 403–422.

[18] Robinson, D.T. and Brown, D.G., (2009). "Evaluating the effects of land-use development policies on ex-urban forest cover: An integrated agent-based GIS approach". International Journal of Geographical Information Science, 23 (9), 1211-1232.

[19] Sengupta, R.R. and Bennett, D.A., (2003). "Agent-based modelling environment for spatial decision support". International Journal of Geographical Information Science, 17 (2), 157-180.

[20] Sheu, J.B., (2007). "An emergency logistics distribution approach for quick response to urgent relief demand in disasters". Transportation Research Part E: Logistics and Transportation Review, 43 (6), 687–709.

[21] Shoham, Y. and Leyton-Brown, K., (2009). "Multiagent Systems: Algorithmic, game-theoretic, and logical foundations", New York: Cambridge University Press.

[22] Uno, K. and Kashiyama, K., (2008). "Development of simulation system for the disaster evacuation based on multi-agent model using GIS". Tsinghua Science & Technology, 13 (1001), 348-353.

[23] Wang, F. and Huang, Q.Y., (2010). "The importance of spatial-temporal issues for case-based reasoning in disaster management". In: Proceedings of the IEEE 18th international conference on geoinformatics. Beijing, China, 1-5.