بهینه سازی تخصیص بار آلودگی در رودخانه
با استفاده از روشهای یادگیری تقویتی

بهزاد شریف، دانشجوی کارشناسی ارشد عمران - محیط زیست دانشگاه علم و صنعت ایران×
سید جمشید موسوی، دانشیار دانشکده مهندسی عمران دانشگاه صنعتی امیرکبیر
b_sharif @ iust.ac.ir ،۰۹۳۲۹۳۹۲۹۴۱

چکیده

استفاده از برنامه ریزی پویای استوکستیک (SDP) در مدلهای بهینه سازی بزرگ مقیاس منابع آب به دلیل نیاز به گسسته سازی متغیرهای حالت و تصمیم و در نتیجه بروز مشکل ابعادی با محدودیتهای جدی مواجه است. روش یادگیری تقویتی (RL) یکی از تکنیکهای پیشرفته مبتنی بر شبیه سازی در حل مسائل تصمیم گیری متوالی در محیط استوکستیک است. در این مقاله، مساله بهینه سازی تخصیص بار آلاینده در رودخانه با استفاده از RL حل شده و کارایی روش با مدل SDP مقایسه گردیده است. نتایج نشان دهنده همگرایی مطلوب روش RL در نیل به جواب بهینه مساله تحت بررسی و سرعت بالاتر آن در مقایسه با روش SDP است

کلید واژه ها: یادگیری تقویتی، برنامه ریزی پویای استوکستیک، تخصیص بار آلودگی در رودخانه

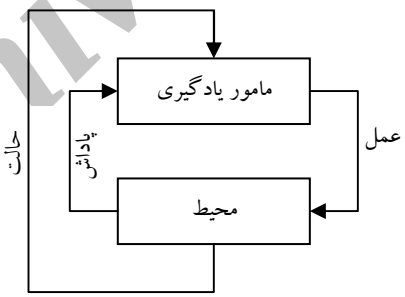
۱- مقدمه

استفاده از روشهای برنامه ریزی پویا (DP)، به دلیل کارآیی آن در حل مسائل غیرخطی و غیر محدب می تواند در سیستمهای رودخانه ای که معادلات روندیابی کیفی و توابع هدف غیرخطی دارند بکار گرفته شود. همچنین عدم قطعیتهای موجود در پارامترهای ورودی مدل نیز می تواند در یک مدل برنامه ریزی پویای استوکستیک لحاظ شود. Lohani & Hee [1]، دبی رودخانه را به عنوان متغیر تصادفی در مدل DP در نظر گرفتند. از آنجا که استفاده از DP نیازی به خطی بودن روابط ندارد، مدلهای با پیچیدگی بالا مانند QUAL-2E را می توان برای محاسبه مقدار اکسیژن محلول در رودخانه ها به عنوان مدل شبیه سازی بکار برد. Takyi & Lence [2] یک مدل زنجیره مارکف غیرایستا را برای بدست آوردن راهبرد مدیریت کیفی آب با استفاده از مقادیر ریسک فصلی توسعه دادند. Mujumdar & Saxena [3] تلاش کردند تا با تلفیق SDP با یک مدل تصمیم گیری فازی، عدم قطعیتهای مربوط به پارامترهای تصادفی و نیز ابهامات موجود در تعریف اهداف را در مدل خود وارد کنند. در این مقاله مساله بهینه سازی احتمالاتی WLA با سیاستهای تصفیه فصلی با استفاده از تکنیک RL فرمولبندی و حل شده و نتایج آن با نتایج مدل SDP مقایسه شده است.

۲- یادگیری تقویتی (RL)

ایده اصلی استفاده از روش یادگیری تقویتی (Reinforcement Learning) یا RL که امروزه در کاربردهای مهندسی رواج یافته است، در واقع برگرفته شده از کار Turdnik (۱۹۱۱) است. وی رفتار حیوانها را از نگاه روانشناسانه مورد بررسی و مطالعه قرار داد. او اعتقاد داشت که اعمالی که توسط یک حیوان در یک موقعیت خاص منجر به نتایج خوبی شده است، تجربه مناسبی برای آن جاندار خواهد شد و اگر آن شرایط مجدداً، نیز تکرار شود آن حیوان تمایل بیشتری به انجام مجدد آن اعمال خواهد داشت. یادگیری تقویتی در واقع پیدا کردن بهترین نوع رفتار در موقعیتهای مختلف در یک سیستم پویا از طریق اندرکنش با محیط اطراف، بدون داشتن یک معلم مشخص است. RL راه حلی برای روند کنترل بهینه است که در آن مأمور یادگیری یا تصمیم گیرنده درصدد یافتن بهترین سیاست است. این سیاست بهینه، در واقع نگاشتی از حالتهای مختلف سیستم به بهترین تصمیمات قابل قبول برای دستیابی به توابع هدف بهینه مشخص برای افق برنامه ریزی است. بنابراین RL بسیار شبیه به روش معمول برنامه ریزی پویا می باشد. در RL سیاست بهینه را می توان به صورت مستقیم بر اساس شبیه سازی بدست آورد. اسامی دیگری برای RL موجود است که از جمله آنها می توان به « برنامه ریزی پویای مبتنی بر شبیه سازی» و «برنامه ریزی پویای عصبی» اشاره کرد.

چهار مولفه اصلی در RL وجود دارد: سیاست، پاداش، تابع ارزش و مدل. سیاست، نگاشتی از حالات به تصمیمهایی است که باید گرفته شود. پاداش، پاسخ فوری محیط به عمل انجام گرفته توسط مأمور یادگیری است. تابع ارزش که برای هر زوج از حالت – عمل (state-action) تعریف می شود، پاداش تجمعی از نقطه شروع RL است. به عبارت دیگر، تابع ارزش بر خلاف تابع پاداش، سود سیستم در هر زوج از حالت – عمل در یک دوره بلند مدت در RL می باشد. مدل در واقع یکی از مؤلفه های اختیاری در RL است که برای تعیین حالت بعد و پاداش بدست آمده بکار می رود. مدل برای حالتی که یادگیری تقویتی قرار است به صورت غیر بهنگام (off-line) انجام شود، اجباری است. شکل شماتیک مولفه های RL و نحوه ارتباط آنها با یکدیگر در زیر نشان داده شده است.



شکل ۱- *شماتیک الگوریتم یادگیری تقویتی*

در بیشتر الگوریتمهای RL، مقدار توابع ارزش همانند روش DP محاسبه می شود و به همین دلیل RL و DP شباهت زیادی به یکدیگر دارند. مقدار تابع ارزش بر اساس معادله بهینگی بلمن از رابطه زیر محاسبه می شود:

$$J^*(i)=\max_{a\in A(i)}\left[\bar{r}(i,a)+\lambda\sum_{j=1}^{|S|}p(i,a,j)J^*(j)\right]\quad \forall i\in S \qquad (۱)$$

که در آن $J^*(i)$، i امین مولفه بردار تابع ارزش مربوط به سیاست بهینه، $A(i)$ مجموعه اعمال قابل انجام در حالت i، $p(i,a,j)$ احتمال انتقال از حالت i به حالت j در نتیجه انجام عمل a، $\bar{r}(i,a)=\sum_{j=1}^{|S|}p(i,a,j)r(i,a,j)$ نشان دهنده مقدار بازگشت فوری مورد انتظار در حالت i ام در صورت انتخاب عمل a است که $r(i,a,j)$ نشان دهنده

بازگشت فوری حاصله از انجام عمل a در حالت i و در نتیجه، رفتن به حالت j است. S، نشان دهنده مجموعه حالتها در زنجیره مارکوفی است و λ نشان دهنده ضریب تنزیل اقتصادی است. برای هر زوج (i,a) به عبارت داخل [ ] در معادله (۱) اصطلاحاً Q-Factor مربوط به آن زوج می گویند.

در DP برای هر مقدار حالت، یک تابع ارزش تخصیص داده می شود، در حالیکه در RL برای هر زوج حالت-عمل، یک تابع ارزش داریم. برای درک بهتر این موضوع فرض کنید که در یک مسئله تصمیم گیری مارکوفی، سه حالت و دو عمل ممکن در هر حالت داریم. در DP، بردار تابع ارزش $\vec{J}^*$ سه عضو همانند ذیل خواهد داشت:

$$\vec{J}^* = \{J^*(1), J^*(2), J^*(3)\}$$

در حالیکه در RL، ۶ مقدار Q-Factor وجود خواهد داشت؛ زیرا ۶ زوج حالت – عمل موجود است. بنابراین اگر $Q(i,a)$ نشان دهنده مقدار Q-Factor مربوط به حالت i و عمل a باشد:

$$\vec{Q} = \{Q(1,1), Q(1,2), Q(2,1), Q(2,2), Q(3,1), Q(3,2)\}$$

برای زوج حالت – عمل (i,a) مقدار Q-Factor متناظر از رابطه زیر محاسبه می شود.

$$Q(i,a) = \sum_{j=1}^{|S|} p(i,a,j)\left[r(i,a,j) + \lambda J^*(j)\right] \qquad (۲)$$

حال، با ترکیب روابط (۱) و (۲) خواهیم داشت:

$$J^*(i) = \max_{a \in A(i)} Q(i,a) \qquad (۳)$$

معادله (۳)، رابطه میان تابع ارزش یک حالت و Q-FACTOR های مرتبط با یک حالت را نشان می دهد. بنابراین اگر Q-FACTOR ها شناخته شده باشند، می توان تابع ارزش یک حالت را از رابطه (۳) بدست آورد. برای مثال، برای حالت i با دو عمل، اگر مقدار Q-FACTOR ها برابر با $Q(i,1) = 95$ و $Q(i,2) = 100$ باشد:

$$J^*(i) = \max\{95,100\} = 100$$

با استفاده از معادله (۳) رابطه (۲) را می توان به صورت زیر نوشت:

$$Q(i,a) = \sum_{j=1}^{|S|} p(i,a,j)\left[r(i,a,j) + \lambda \max_{b \in A(j)} Q(j,b)\right] \qquad (۴)$$

رابطه (۴) در واقع به عنوان نسخه Q-FACTOR رابطه بهینگی بلمن قابل تعبیر است.

محاسبه مقادیر Q-FACTOR روش تکرار ارزش (Value Iteration Method)

الگوریتمی که در ذیل ارائه می شود معادل الگوریتم تکرار ارزش متداولی است که در DP مورد استفاده قرار می گیرد. این الگوریتم شامل مراحل زیر است.

گام اول: مقدار شماره گام زمانی k ام را برابر ۱ قرار داده و بردار دلخواه $\vec{Q}_0$ انتخاب می شود. برای مثال، برای تمام $i \in S$ و $a \in A(i)$:

$$Q^0(i,a) = 0$$

مقدار ε (معیار توقف) بزرگتر از صفر قرار داده می شود.

گام دوم: برای هر $i \in S$ تابع ارزش به شکل زیر محاسبه می شود:

$$Q^{k+1}(i) \leftarrow \sum_{j=1}^{|S|} p(i,a,j)\left[r(i,a,j) + \lambda \max_{b \in A(j)} Q^k(j,b)\right]$$

گام سوم: برای هر $i \in S$ تابع ارزش بهینه به شکل زیر محاسبه می شود:

$$J^{k+1}(i) = \max_{a \in A(i)} Q^{k+1}(i,a), \quad J^k(i) = \max_{a \in A(i)} Q^{k+1}(i,a)$$

در ادامه اگر $\left\|\vec{J}^{k+1}-\vec{J}^{k}\right\|<\varepsilon(1-\lambda)/2\lambda$ به گام ۴ رفته ودر غیر این صورت k به اندازه یک واحد افزایش می یابد و کنترل به گام ۲ برمی گردد.

گام چهارم: برای تمام $i\in S$ تصمیم بهینه بدین صورت محاسبه می شود: $d(i)\in \arg\max_{b\in A(j)} Q(i,b)$

$\hat{d}$ سیاست ε – بهینه نامیده می شود. در صورت ارضای رابطه شامل عبارت ε در گام سوم، برای تمامی حالتها، الگوریتم متوقف می گردد. معادله گام ۲ از رابطه(۴) استخراج شده است. تشخیص معادل بودن این الگوریتم با روش تکرار ارزش معمول چندان دشوار نیست. به جای برآورد کردن مقدار تابع ارزش، این الگوریتم Q-FACTOR ها را برآورد می کند. در RL نیز Q-FACTOR ها برآورد می شوند ولی الگوریتم به روز رسانی معادله کمی متفاوت با رابطه ای است که در گام ۲ بیان شد.

الگوریتم رابینز – مونرو

الگوریتم رابینز–مونرو، الگوریتمی قدیمی از دهه پنجاه میلادی است که بوسیله آن میانگین یک متغیر تصادفی از روی نمونه های تولید شده از آن برآورد می شود. میانگین یک متغیر تصادفی را می توان با میانگین گیری مستقیم بدست آورد. فرض کنید i امین نمونه از متغیر تصادفی تولید شده X، s^i باشد و مقدار امید ریاضی این نمونه ها $E(X)$ باشد. مقدار میانگین حاصله از رابطه $\sum_{i=1}^{n} s^i \Big/ n$ با رفتن n به سمت بی نهایت با احتمال ۱ به مقدار واقعی میانگین همگرا می شود. (این مسئله از قانون مهم اعداد بزرگ به دست می آید)، به عبارت دیگر، با احتمال ۱، $E(X)=\lim_{n\to\infty}\sum_{i=1}^{n} s^i \Big/ n$.

می توان از این رابطه، الگوریتم رابینز–مونرو را استخراج کرد. فرض کنید مقدار برآورد شده X در n امین تکرار – بعد از تولید n نمونه – X^n باشد. بنابراین $X^n=\sum_{i=1}^{n} s^i \Big/ n$. در نتیجه:

$$X^{n+1}=\frac{\sum_{i=1}^{n+1}s^i}{n+1}=\frac{\sum_{i=1}^{n}s^i+s^{n+1}}{n+1}=\frac{X^n n+s^{n+1}}{n+1}=\frac{X^n n+X^n-X^n+s^{n+1}}{n+1}=\frac{X^n(n+1)}{n+1}-\frac{X^n}{n+1}+\frac{s^{n+1}}{n+1}$$

$$=X^n-\frac{X^n}{n+1}+\frac{s^{n+1}}{n+1}=(1-\alpha^{n+1})X^n+\alpha^{n+1}s^{n+1}\quad if\quad \alpha^{n+1}=1/n+1$$

در نهایت خواهیم داشت:

$$X^{n+1}=(1-\alpha^{n+1})X^n+\alpha^{n+1}s^{n+1} \qquad (۵)$$

اگر α^{n+1} برابر با $1/(n+1)$ باشد، این روش معادل میانگین گیری ساده خواهد بود.

الگوریتم رابینز–مونرو و برآورد Q-FACTOR ها

الگوریتم رابینز–مونرو را می توان برای برآورد Q-FACTOR ها بکار برد. هر Q-FACTOR در واقع میانگین یک متغیر تصادفی است. معادله بهینگی بلمن به شکل زیر نیز قابل بیان است:

$$Q(i,a)=\sum_{j=1}^{S}p(i,a,j)\left[r(i,a,j)+\lambda\max_{b\in A(j)}Q(j,b)\right]=E\left[r(i,a,j)+\lambda\max_{b\in A(j)}Q(j,b)\right] \qquad (۶)$$

که E عملگر امید ریاضی و مقدار داخل کروشه در معادله (۶) یک متغیر تصادفی است. بنابراین اگر نمونه هایی از متغیر تصادفی از طریق شبیه سازی به دست آورده شود، می توان به جای بهره گیری از معادله (۶) برای برآورد Q-FACTOR ها (همانطور که در نسخه Q-FACTOR روش تکرار ارزش ها آمده است) می توان از الگوی رابینز–مونرو برای ارزیابی

Q-FACTOR ها استفاده کرد. با استفاده از الگوریتم رابینز-مونرو (رابطه ۵)، رابطه (۶) برای هر زوج حالت-عمل به صورت زیر خواهد بود:

$$Q^{n+1}(i,a) \leftarrow (1-\alpha^{n+1})Q^n(i,a) + \alpha^{n+1}\left[r(i,a,j) + \lambda \max_{b\in A(j)} Q^n(j,b)\right] \qquad (۷)$$

جالب ترین مسئله در مورد این الگوریتم این است که در این روش احتمال انتقال حالت وجود ندارد و تنها به یک شبیه ساز برای سیستم نیاز است. مکانیسم نشان داده شده باعث جلوگیری از تشکیل ماتریس احتمال انتقال در RL می شود. نیاز به برآورد این ماتریس در مدلهای بزرگ مقیاس با تعداد متغیرهای تصادفی قابل ملاحظه، تحت عنوان بلای مدلسازی (curse of modeling) شناخته می شود که RL این مشکل را حل می کند.

روش تکرار ارزش در RL

فرض کنید که شبیه ساز عمل a را در مرحله i انتخاب کند و سیستم در نتیجه عمل i به حالت j برود. در طول دوره زمانی که شبیه ساز از حالت i به j می رود، الگوریتم RL از اطلاعات درون شبیه ساز استفاده می کند. این اطلاعات شامل $r(i,a,j)$ است که بازگشت حاصله از رفتن از حالت i به حالت j در نتیجه انجام عمل a است. وقتی شبیه ساز به حالت j می رسد از $r(i,a,j)$ برای به روز رسانی مقدار $Q(i,a)$ اقدام می کند. بنابراین به روز رسانی، پس از انتقال کامل از حالت i به حالت j رخ می دهد. با توجه به مطالب ذکر شده، روش RL شامل مراحل زیر است.

گام ۱: شماره تکرار الگوریتم (k=1) و سپس تمام مقادیر Q-FACTOR را برابر صفر قرار می دهیم. به عبارت دیگر برای همه (l,u) که $l \in S$ و $u \in A$ قرار می دهیم: $Q(l,u) \leftarrow 0$

گام ۲: فرض کنید سیستم در حالت i باشد. عمل a را با احتمال $1/A(i)$ (که $A(i)$ مجموعه تصمیمات ممکن در حالت i ام است) انتخاب کرده و سیستم شبیه سازی می شود.

گام ۳: فرض کنید که حالت بعدی سیستم برابر با j باشد. $r(i,a,j)$ بازگشت فوری بدست آمده در انتقال از حالت i به حالت j، تحت عمل a می باشد که توسط شبیه ساز به دست می آید. در این شرایط $V(i,a)$ که بیانگر تعداد دفعاتی است که هر زوج حالت-عمل (i,a) مورد امتحان قرار می گیرد را یک واحد افزایش می دهیم. این مقدار همان n در رابطه (۷) می باشد که به فاکتور برخورد (Visit Factor) موسوم است. k را به اندازه یک واحد افزایش داده و مقدار $\alpha = 1/V(i,a)$ محاسبه می شود.

گام ۴: $Q(i,a)$ با استفاده از معادله زیر به روز رسانی می گردد:

$$Q(i,a) \leftarrow (1-\alpha)Q(i,a) + \alpha\left[r(i,a,j) + \lambda \max_{b\in A(j)} Q(j,b)\right]$$

گام ۵: اگر $k < k_{\max}$، $i \leftarrow j$ و به گام ۲ رفته، وگرنه گام ۶ انجام می شود.

گام ۶: برای هر $l \in S$ انتخاب می کنیم: $d(l) \in \arg\max_{b\in A(l)} Q(l,b)$

سیاست (راه حل) بدست آمده توسط الگوریتم، بردار $\hat{d}$ خواهد بود. گامهای مذکور تا $k = k_{\max}$ تکرار می شود.

روشهای انتخاب عمل (Action Selection Methods)

در بخش قبل و پس از توضیح روش الگوریتم رابینز – مونرو در این خصوص توضیح داده شد که برای میانگین گیری، تمام اعمال با احتمال مساوی شانس انتخاب شدن را دارند. اگر تمام اعمال برای تمام حالتهای ممکن به تعداد دفعات زیادی تکرار شود، چند مشکل پدید می آید. اول آن که اگر بخواهیم تمام حالتها و اعمال به تعداد زیاد (مناسب) تکرار شوند تا میانگین گیری به واقعیت نزدیک باشد، تعداد گامهای زمانی زیادی در شبیه سازی باید انجام شود. دوم اینکه با این

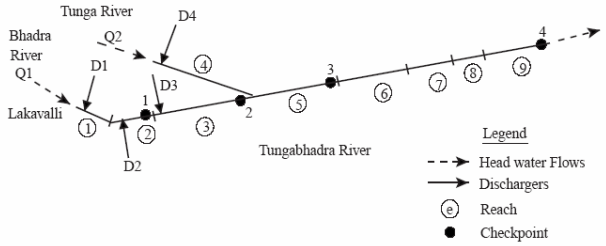
کار، عملاً روش انتخاب سیاست بهینه بر مبنای تکرار بلند مدت شبیه سازی خواهد بود و هیچ تلاشی در جهت هوشمند کردن انتخاب اعمال مناسب در هر حالت وجود نخواهد داشت. یکی از ساده ترین روشها برای هوشمند کردن انتخاب اعمال مناسب، انتخاب کردن عملی است که دارای بیشترین میزان ارزش برآورد شده تا به حال است. یعنی به صورت کاملاً « حریصانه » (greedy)، در t امین گام زمانی، عمل a^* به نحوی انتخاب شود که $Q_t(a^*) = \max_a Q_t(a)$. در این روش همواره از اطلاعات کنونی برای حداکثر کردن پاداش فوری استفاده می شود و توجهی به اعمالی که دارای ارزش کنونی کمتری هستند و ممکن است واقعاً دارای ارزش تجمعی بیشتری باشند نمی شود. روش جایگزین برای روش «حریصانه» این است که در بیشتر اوقات به صورت حریصانه اعمال را انتخاب کنیم، ولی گاهی اوقات، با احتمالی کوچک، ε، عملی دیگر را به صورت تصادفی و بدون توجه به توابع ارزش محاسبه شده انتخاب کنیم. این روشها که بسیار نزدیک به روشهای حریصانه هستند را روشهای ε –حریصانه $(\varepsilon - greedy)$ می نامند. مزیت این روش این است که با افزایش تعداد تکرارها، هر از چند گاهی اعمال دیگری نیز غیر از اعمال حریصانه انتخاب می شوند و این به همگرایی ارزش تخمین زده شده به ارزش واقعی کمک می کند. زیرا ممکن است این اعمال که در کوتاه مدت دارای ارزش کمتری هستند، در ادامه ارزشی بیشتری از اعمال حریصانه داشته باشند.

ارزیابی عملکرد

برای تعیین کیفیت راه حل ارائه شده توسط مسئله، باید بعد از بدست آوردن سیاستهای بهینه، مسئله را شبیه سازی نمود. این شبیه سازی، با استفاده از سیاست یاد گرفته شده در آخرین گام الگوریتم انجام می شود. برای پیدا کردن مقدار تابع ارزش حالت i، باید شبیه سازی را از حالت i شروع کرد و پاداشهای بدست آمده در طول یک دوره شبیه سازی را محاسبه نمود. این کار باید در چندین تکرار انجام گردیده و از جوابها میانگین گیری شود.

۳- مطالعه موردی

برای بررسی نحوه بکارگیری الگوریتم RL در مسئله تخصیص بار آلودگی، از مطالعه موردی که توسط Mujumdar و Saxena (۲۰۰۴) انجام شده است، استفاده شده است. ناحیه مورد مطالعه آنها در شکل (۲) نشان داده شده است. همانگونه که از شکل مشخص است، این مسئله شامل بهینه سازی تخصیص بار ۴ واحد آلاینده (D1,D2,D3,D4) در یک سیستم رودخانه ای است که از به هم پیوستن دو رودخانه Tunga و Bhadra در کشور هند تشکیل شده است.



شکل ۲-*شماتیک ناحیه مورد مطالعه برای بهینه سازی تخصیص بار آلودگی*

برای بررسی وضعیت تغییرات فصلی این سیستم، هر سال از لحاظ شرایط اقلیمی به ۳ فصل تقسیم شده است. سیستم رودخانه ای به ۹ بازه تقسیم شده که در هر بازه شرایط هندسی رودخانه ثابت فرض شده است. ۴ نقطه کنترل (Checkpoint) برای کنترل آلودگی در طول سیستم رودخانه ای در نظر گرفته می شود. متغیرهای حالت این مسئله، بردار میزان کمبود اکسیژن در چهار نقطه کنترل آلودگی و متغیرهای تصمیم این مسئله، برابر بردار میزان تصفیه در واحدهای آلاینده فرض شده است. متغیرهای حالت در هر نقطه کنترل به ۶ کلاس و متغیرهای تصمیم به ۹ کلاس تقسیم شده اند. برای پارامتر دبی ورودی در هر کدام از سرشاخه های رودخانه نیز در هر فصل ۴ کلاس تعریف شده و برای مسئله SDP

ماتریس احتمال انتقال آن ساخته شده است. بنابراین با مسئله ای با ابعاد بسیار بزرگ روبرو هستیم که حل آن توسط SDP مستلزم هزینه وقت و حافظه زیادی می باشد. برای شبیه سازی سیستم از مدل معروف استریتر-فلپس برای روندیابی میزان کمبود اکسیژن در طول رودخانه استفاده شده است.

در هر فصل، بر اساس میزان کمبود اکسیژن محلول در نقاط کنترل و دبی جریان در رودخانه ها، شبیه سازی انجام می شود و مقادیر کمبود اکسیژن محلول در نقاط کنترل در انتهای آن فصل (ابتدای فصل بعد) به دست می آید. در این انتقال حالت، معیار عملکرد سیستم، از یک مدل تصمیم گیری فازی بدست می آید که در برگیرنده دو هدف در تضاد با یکدیگر است. هدف اول مربوط به سازمانهای محیط زیست است که تمایل به وضعیت کیفی بهتر دارند و هدف دوم مربوط به تخلیه کنندگان آلاینده ها است که تمایل به تصفیه کمتری دارند. این اهداف به صورت فازی با توابع عضویت خطی تعریف شده و میزان معیار عملکرد سیستم، حداکثر میزان ارضای این اهداف قرار داده می شود. تابع هدف مسئله و قیود مربوط به اهداف فازی در ذیل آمده است.

$$\text{maximize } \lambda$$

Subject to:

$$\lambda^* = \max_{y \in Z} [\mu_Z(Y)]$$

$$\mu_z(Y) = \overset{Minimum}{w,c,d} [\mu E_{wc}(a_{wc}), \mu F_{wd}(x_{wd})]$$

$$\mu E_{wc}(a_{wc}) = \begin{cases} 1 & a_{wc} < a_{wc}^D \\ \left[a_{wc}^H - a_{wc} / a_{wc}^H - a_{wc}^D\right]^{\alpha_{wc}} & a_{wc}^D \le a_{wc} \le a_{wc}^H \\ 0 & a_{wc} > a_{wc}^H \end{cases}$$

$$\mu F_{wd}(x_{wd}) = \begin{cases} 1 & x_{wd} < x_{wd}^{As} \\ \left[x_{wd}^M - x_{wdn} / x_{wd}^M - x_{wd}^{As}\right]^{\beta_{wd}} & x_{wd}^M \le x_{wd} \le x_{wd}^M \\ 0 & x_{wd} > x_{wd}^M \end{cases}$$

در روابط بالا، λ معیار عملکرد سیستم حاصل از مدل تصمیم گیری فازی، Y فضای تصمیم گیری، w شاخص کیفی (در اینجا اکسیژن محلول)، c نقطه کنترل، d واحد آلوده کننده، E و F به ترتیب اهداف سازمانهای محیط زیست و واحدهای آلاینده، a میزان غلظت آلاینده، a^H و a^D مقادیر حداکثر غلظت مجاز و غلظت مطلوب آلاینده، x میزان تصفیه، x^M و x^{As} به ترتیب مقادیر تصفیه حداکثر و مطلوب واحدهای آلاینده و αو β، پارامترهای مربوط به شکل توابع عضویت می باشند. توضیحات بیشتر در مورد سایر اطلاعات و جزئیات مسئله در [۳] ارایه شده است.

۴- بررسی و تحلیل نتایج

از مدل SDP و نیز مدل RL برای بدست آوردن سیاست بهینه استفاده شد و پس از هر اجرا، شبیه سازی کیفی سیستم رودخانه با توجه به سیاست بهینه تصفیه انجام شد و میزان تابع هدف حاصله بدست آمد. برای انتخاب عمل از سه روش انتخاب تصادفی اعمال (الگوریتم اصلی رابینز-مونرو)، انتخاب اعمال حریصانه و انتخاب اعمال به صورت ε-حریصانه، با مقدار ε برابر ۰/۱ ، مورد بررسی قرار گرفت. مدلهای RL چندین بار اجرا شد و نتایج از میانگین گیری جوابها حاصل گردید. این نتایج در جدول (۱) خلاصه شده است. نتایج نشان دهنده این است که روش RL می تواند در زمانی بسیار سریع تر از روش SDP، به جوابهای نزدیک به جواب بهینه دست پیدا کند. دلیل این مسئله آن است که تعداد بسیار کمی از ترکیبات موجود در بردار حالت سیستم در شبیه سازی سیاستهای بهینه در تخصیص بار آلودگی رودخانه مصداق می یابند و لذا تعیین کردن یا نکردن سیاست بهینه برای این حالتها ضرورت زیادی ندارد. انتخاب کردن حریصانه اعمال، تنها در

حالتی می تواند جوابگو باشد که شبیه سازی از یکی از حالتهایی که در عمل احتمال رخ دادن آنها زیاد است شروع شود. در غیر این صورت، الگوریتم RL در حلقه ای گیر می افتد که جوابی به دور از جواب بهینه خواهد داشت. روش انتخاب تصادفی اعمال، این مشکل را از بین می برد، ولی به دلیل ابعاد بزرگ مسئله، پراکندگی جوابها در اجراهای مختلف کمی بیشتر می گردد. در روش انتخاب اعمال ε -حریصانه، مشکل گیر افتادن مسئله در حلقه تکراری حل می شود، زیرا در بعضی اوقات، اعمالی غیر از اعمال حریصانه انتخاب می شوند. اما به دلیل اینکه در اکثر مواقع، بهترین اعمال انتخاب می شوند، پراکندگی کمتری در جوابهای تابع هدف بهینه به چشم می خورد.

جدول ۱- *مقایسه نتایج حاصل از بکارگیری روشهای RL و SDP در محاسبه تابع هدف مسئله تخصیص بار آلودگی رودخانه بهادرا و تونگا در هندوستان*

روش	نوع انتخاب عمل	میانگین تابع هدف بهینه	انحراف معیار	حدود زمان محاسباتی
SDP	-	۰/۳۴۸۹	۰/۰۰۲۲	۱۲ ساعت
RL	حریصانه	۰/۲۹۹۵	۰/۰۰۲۵ *	۵۹ ثانیه
RL	ε - حریصانه	۰/۳۴۴۶	۰/۰۰۴۷	۶۲ ثانیه
RL	تصادفی	۰/۳۲۱۵	۰/۰۰۸۱	۵۹ ثانیه

*: *در صورتیکه حالت اولیه سیستم را تغییر ندهیم*

۵- خلاصه و نتیجه گیری

روشهای بهینه سازی مبتنی بر شبیه سازی در کارهای اخیر مورد توجه بسیاری قرار گرفته است. یکی از مناسب ترین روشهای حل مسائل غیرخطی دارای عدم قطعیت، روش برنامه ریزی پویای استوکستیک است. روشهای یادگیری تقویتی که گاهی از آنها به عنوان روش برنامه ریزی پویای مبتنی بر شبیه سازی یاد می شود، بدون نیاز به ماتریس احتمال انتقال و با استفاده از یک شبیه ساز که داده ها در آن تولید شده و شبیه سازی به صورت مستقیم و رو به جلو انجام می شود، با استفاده از تجربیات مثبت و منفی حاصله، مقدار توابع ارزش را برای حالتها و تصمیمهای مختلف به روز می کند. مهمترین مزیت روش یادگیری تقویتی، دستیابی به جواب تقریباً بهینه، بدون نیاز به محاسبه کردن تابع ارزش برای تک تک زوجهای حالت-تصمیم است. در این روش تنها حالتهایی که در واقعیت بیشتر مشاهده می شوند، در محاسبات و به روز رسانی تابع ارزش در نظر گرفته می شوند. بخصوص استفاده از روشهای ε -حریصانه، با همگرایی سریع، جوابهایی بسیار نزدیک به جواب بهینه حاصل از مدل معمول SDP فراهم می کند. نتایج استفاده از این روشها در بهینه سازی استوکستیک تخصیص بار آلاینده در سیستم رودخانه تحت بررسی در این مطالعه امیدوار کننده میباشد. بنابراین این روشها در حل مسائل بزرگ مقیاس که مدلهای SDP در آنها دچار مشکل بعد (Curse Of Dimensionality) می باشند، جایگزین مناسبی خواهند بود.

۶- مراجع

[1] Lohani B.N., Hee K.B. (1983), A CCDP model for water quality management in Hisntein River in Taiwan, Intl Journal of Water Resource Dev., 1, pp 91-114

[2] Takyi A.K, Lence B.J., "Markov chain model for seasonal-water quality management", Journal of Water Resourc. Pla, Vol.121,NO.2 , March/April, 1995 (ASCE),pp 144-157

[3] Mujumdar P.P, Saxena P, "A strochastic dynamic programming model for stream water quality management", Journal of Sadhana, Vol.29, Part5, October,2004, pp.1-22.

[4] Sutton R.S., and Barto A.G. 1998, Reinforcement Learning. The MIT press, Cambridge, Massachusetts.

[5] Gosavi A. 2003, Simulation-based optimization: parametric optimization techniques and reinforcement learning. Kluwer academic publisher, Norwel Massachusetts.