

## یک الگوریتم ترکیبی برای حل مسائل ارضای محدودیت

محمدرضا میبیدی  
دانشگاه صنعتی امیرکبیر تهران  
mmeybodi@ce.aut.ac.ir

عبدالرضا حاتم‌لو  
دانشگاه آزاد اسلامی واحد خوی  
reza\_hatamloo@yahoo.com

آنها را بدست آورد. از اینرو تلاشهای زیادی برای طراحی و ارائه الگوریتم های کارآمد برای جستجوی جواب های این مسائل صورت گرفته و می گیرد. تاکنون الگوریتم های مختلف و زیادی برای حل مسائل ارضای محدودیت ارائه شده اند که هر کدام از آنها معایب و مزایایی دارند. از یک دیدگاه الگوریتم های موجود به دو دسته اصلی کامل و ناکامل دسته بندی می شوند.

الگوریتم های کامل [1-3] کل فضای مسئله را بطور سیستماتیک و به ترتیب جستجو می کنند. این الگوریتم ها دقیق و کامل هستند یعنی اگر مسئله ای دارای جواب باشد این الگوریتم ها تضمین می کنند که جواب را بیابند علاوه در صورت نیاز تمام جواب های مسئله را می یابند. این الگوریتم ها از یک جواب تهی شروع می کنند یعنی حالتی که در آن هیچ یک از متغیرها برچسب دهی نشده اند و این جواب را مرحله به مرحله به سوی یک جواب صحیح و کامل نزدیک می کنند. در هر مرحله ای هم که ناسازگاری پیش می آید عقبگرد انجام می گیرد. بزرگترین ایراد و مشکل این دسته از الگوریتم ها این است که اگر یک انتساب و برچسب دهی اشتباهی در ابتدای عمل جستجو صورت گیرد این اشتباه ممکن است تا مراحل زیادی تشخیص داده نشود و بخش زیادی از فضای مسئله بدون اینکه هیچ جوابی در آن وجود داشته باشد کاوش شود و در نتیجه زمان اجرای الگوریتم افزایش یافته و کارایی آن پایین بیاید. همچنین در کاربردهایی که زمان برای ما مهم است و فقط یک جواب برای مسئله کافی است این الگوریتم ها کارایی ندارند چون زمان آنها خیلی بالا است.

در مقابل الگوریتم های ناکامل [5-7] برخلاف الگوریتم های بالا کامل و صحیح نیستند یعنی اولاً تمام جوابهای مسئله را پیدا نمی کنند و بیشتر برای مسائلی که تنها یک جواب برای آنها لازم است بکار می روند و ثانیاً هیچ تضمینی هم برای پیدا کردن جواب نمی دهند. این الگوریتم ها از یک حالتی که تمام متغیرها در آن برچسب دهی شده اند شروع می کنند که البته در این حالت ممکن است تعدادی از انتساب ها نادرست باشند و بعضی از محدودیت های مسئله را نقض کنند. الگوریتم های غیرکامل مدام از یک حالت به یک حالت دیگر حرکت می کنند تا به جواب نزدیکتر شوند. به بیان دیگر در این الگوریتم ها در هر لحظه مجموعه ای از برچسب ها سازگار و مجموعه ای از برچسب ها ناسازگار هستند برخلاف الگوریتم های کامل که در هر لحظه تمام برچسب ها

**چکیده:** بسیاری از مسائل مطرح در هوش مصنوعی را می توان بصورت یک مسئله ارضای محدودیت تعریف کرد و سپس با استفاده از الگوریتم هایی که برای حل این مسائل ارائه شده اند جواب آنها را پیدا کرد. چون فضای حالات مسائل ارضای محدودیت خیلی وسیع است و زمان زیادی برای جستجوی آن لازم است تلاش های زیادی برای طراحی و ارائه الگوریتم های کارا برای حل اینگونه مسائل انجام می گیرد. در این مقاله یک الگوریتم ترکیبی برای حل مسائل ارضای محدودیت ارائه می کنیم که ترکیبی از الگوریتم های کامل و ناکامل است. امتیاز این الگوریتم پیدا کردن جواب های جزئی صحیح در زمان کوتاه برای مسائلی می باشد که یافتن جواب کامل برای آنها در زمان قابل قبول امکانپذیر نمی باشد. الگوریتم ترکیبی را برای مسئله مشهور N-Queens پیاده سازی کرده ایم و نتایج آن را با الگوریتم های موجود مقایسه کرده ایم.

**واژه های کلیدی:** هوش مصنوعی، مسائل ارضای محدودیت، الگوریتم های جستجو.

### ۱- مقدمه

بسیاری از مسائل مطرح در زمینه هوش مصنوعی، شبکه های کامپیوتری، پایگاه داده و حوزه های مختلف علم کامپیوتر را می توان به صورت یک مسئله ارضای محدودیت در نظر گرفت. مثال هایی از این نوع مسائل عبارتند از: زمانبندی، طراحی مدارات الکترونیکی و دیجیتال، پردازش زبانهای بشری، بازی های کامپیوتری و پردازش تصویر.

یک مسئله ارضای محدودیت متناهی تشکیل شده است از مجموعه ای از متغیرها که هر یک از این متغیرها دامنه محدودی از مقادیر را می توانند داشته باشند. همچنین محدودیت هایی بر روی مقادیری که متغیرهای مسئله می توانند بطور همزمان و در ارتباط با همدیگر داشته باشند تعریف می شوند. جواب یک مسئله ارضای محدودیت عبارت است از انتساب یک مقدار برای هر یک از متغیرهای مسئله از دامنه آنها بطوریکه تمام محدودیتهای تعریف شده در مسئله رعایت شده باشند [1-4]. مشکلی که در حل مسائل ارضای محدودیت وجود دارد این است که فضای حالات این مسائل خیلی وسیع است و جستجوی این فضای بزرگ برای پیدا کردن جواب های مسئله مستلزم صرف زمان طولانی می باشد بطوریکه بعضی از این مسائل هفته ها زمان نیاز دارند تا جواب

شود. در طول جستجو نگهداری سازگاری جواب از طریق حذف برچسب های برخورد دار یعنی برچسب هایی که باعث نقض محدودیت های مسئله می شوند حاصل می شود. الگوریتم زمانی پایان می یابد که تمام متغیرهای مسئله با موفقیت برچسب دهی شوند یا اینکه ماکزیمم تعداد تکرارهای در نظر گرفته شده برای پیدا کردن جواب انجام گرفته باشد. این الگوریتم دو تابع اصلی دارد. در هر تکرار یکی از آنها یک متغیر بدون مقدار را برای برچسب دهی انتخاب می کند. چندین معیار برای اینکار وجود دارد. در این الگوریتم از قانون شکست در ابتدا [1,2] برای گزینش متغیر بعدی استفاده شده است. یعنی در هر مرحله متغیری برای برچسب دهی انتخاب می شود که در محدودیت های زیادی از مسئله شرکت داشته باشد و یا اندازه دامنه مقادیر آن از بقیه متغیرها کوچکتر باشد. بعد از انتخاب یک متغیر تابع دیگری در این الگوریتم مقداری را برای آن متغیر از دامنه اش انتخاب می کند. برای انتخاب مقدار برای یک متغیر هم روش هایی وجود دارد که در اینجا قانون بهترین مقدار [1,2] به کار برده شده است. یعنی تابع مقداری را برای یک متغیر انتخاب می کند که کمترین احتمال برخورد با متغیرهای دیگر را در ادامه خواهد داشت به بیان دیگر مقدار انتخابی باید متغیرهای دیگر را تا حد امکان محدود نکند. شبه کد الگوریتم ترکیبی بصورت زیر می باشد:

```
procedure hybrid(unlabeled, answer, max_repeat)
/* unlabeled is a set of un-labeled variables and answer is a
incomplete answer (empty at the start)*/
  repeats=0;
  while unlabeled not empty & repeats<max_repeat
    repeats++;
    variable = Variable_Chooser(unlabeled, answer);
    unlabeled -= variable;
    value = Value_Chooser(variable, answer);
    unlabeled += label(answer, variable, value)
    // label the variable and return conflict variables
  end while;
  return answer;
end hybrid
```

### ۳- نتایج پیاده سازی

زمان لازم برای حل مسئله N-وزیر توسط الگوریتم ترکیبی در مقایسه با الگوریتم های کامل و ناکامل در جدول (۱) نشان داده شده است. مسئله N-وزیر عبارت است از قرار دادن N عدد وزیر در یک صفحه شطرنجی  $N \times N$  بطوریکه هیچ یک از وزیرها همدیگر را تهدید نکنند. الگوریتم کاملی که در اینجا استفاده شده است سریعترین الگوریتم از این نوع بنام الگوریتم آینده نگر کامل [1,2] است. الگوریتم ناکامل بکار رفته هم سریعترین الگوریتم از این نوع بنام الگوریتم برخورد-کمینه [1,2] است. نتایج ارائه شده بر روی یک کامپیوتر با پردازنده پنتیوم IV و اندازه حافظه اصلی 512MB و در روی سیستم عامل ویندوز XP بدست آمده اند.

سازگاران و در موقع بروز اولین ناسازگاری یا نقض محدودیت عقبگرد انجام می گیرد تا دوباره به حالت سازگار برگشت کنند. این الگوریتم ها همچنان که از نامشان پیداست برای پیدا کردن جواب مسئله کل فضای آن را جستجو نمی کنند بلکه عمل جستجو را بصورت محلی انجام می دهند در نتیجه هیچ تضمینی برای پیدا کردن جواب نمی دهند ولی چون بخش محدودی از فضای مسئله را جستجو می کنند در صورت وجود جواب در آن بخش در زمان خیلی کمتری آن را پیدا می کنند. در مقایسه این دو دسته از الگوریتم ها می توان گفت که اگر یک مسئله ارضای محدودیت دارای جواب باشد الگوریتم های کامل تضمین می کنند که جواب را بیابند و هر جواب جزئی تولید شده توسط این الگوریتم ها در حالت سازگار خواهد بود یعنی تمام متغیرهای برچسب دهی شده تمام محدودیت های مسئله را ارضا خواهند کرد ولی زمان پیدا کردن جواب خیلی طولانی می باشد. در مقابل الگوریتم های غیرکامل هیچ تضمینی نمی کنند که جواب را بیابند و حالت های میانی تولید شده توسط اینها ناسازگار خواهد بود به این معنی که بعضی از متغیرهای برچسب دهی شده فعلی بعضی از محدودیت های مسئله را نقض می کنند ولی به علت استفاده از تکنیک های غیرکامل جواب را در زمان خیلی کمتری نسبت به الگوریتم های کامل پیدا می کنند.

### ۲- الگوریتم ترکیبی

الگوریتم ترکیبی که ما در این مقاله ارائه می کنیم از قابلیت های هر دو دسته از الگوریتم های کامل و ناکامل استفاده می کند بدین صورت که همانند الگوریتم های کامل تمام حالت های میانی تولید شده توسط آن سازگار خواهند بود و جواب های جزئی تولید شده بوسیله این الگوریتم تمامی محدودیت های مسئله را ارضا می کند. از طرفی همانند الگوریتم های غیرکامل فضای مسائل ارضای محدودیت را بصورت محلی کاوش می کند یعنی تغییرات حالت بعدی مسئله نسبت به حالت فعلی خیلی ناچیز است و در نتیجه جواب را در زمان خیلی کمتری پیدا می کند.

این الگوریتم به صورت تکراری انجام می شود. در ابتدا الگوریتم همانند الگوریتم های کامل با یک جواب تهی شروع می کند یعنی حالتی که هیچ یک از متغیرهای مسئله برچسب دهی نشده اند. سپس در هر تکرار یک متغیر بدون برچسب انتخاب می شود. بعد از انتخاب یک متغیر، مقداری از دامنه آن انتخاب می شود و به متغیر منتسب می شود. برچسب تولید شده جدید ممکن است باعث نقض محدودیتی شود یعنی با مقادیر متغیرهایی که قبلاً برچسب دهی شده اند ناسازگار باشد. چنین متغیرهایی برخلاف عملکرد الگوریتم های غیرکامل از جواب جزئی فعلی حذف می شوند و به لیست متغیرهای بدون برچسب اضافه می شوند و دوباره الگوریتم تکرار می شود. این الگوریتم در تکرارهای متوالی از یک جواب جزئی ولی سازگار به یک جواب جزئی و سازگار دیگر حرکت می کند و به جواب کامل و صحیح مسئله نزدیکتر می

#### ۴- نتیجه گیری

در این مقاله ما یک الگوریتم برای حل مسائل ارضای محدودیت ارائه کردیم که ترکیبی از الگوریتم های کامل و ناکامل است. انگیزه اصلی برای ارائه چنین الگوریتمی حل مسائل ارضای محدودیت بزرگ و پیچیده ای است که تعداد متغیرها و محدودیت های آنها زیاد است و یافتن جواب کامل صحیح برای آنها غیرممکن و یا زمانبر است. سرعت این الگوریتم همچنانکه در بالا نشان داده شد نزدیک سرعت الگوریتم های ناکامل می باشد که سریعترین الگوریتم ها برای حل مسائل ارضای محدودیت می باشند. از طرفی همانند الگوریتم های کامل هر حالت میانی تولید شده توسط الگوریتم ترکیبی در حالت سازگار قرار دارد یعنی متغیرهای برچسب دهی شده در هر حالت تمامی محدودیت های مسئله را ارضا می کنند. بنابراین به نظر می رسد در مسائل بزرگ و پیچیده ای که فضای آنها وسیع است و پیدا کردن جواب کامل نهایی در زمان قابل قبول امکانپذیر نمی باشد می توان با استفاده از الگوریتم ترکیبی جواب جزئی مسئله را در زمان قابل قبولی بدست آورد. زیرا سرعت الگوریتم های کامل برای مسائل بزرگ خیلی پایین است و الگوریتم های ناکامل هم از تولید جواب جزئی صحیح عاجز هستند.

الگوریتم ترکیبی ارائه شده در این مقاله می تواند برای مسائل دیگری نیز بکار برده شود و نتایج آن با الگوریتم های موجود مقایسه شود. از جمله کاربردهایی که به نظر می رسد الگوریتم ترکیبی گزینه بهتری برای آنها باشد مسائل پیچیده ای از قبیل زمانبندی و بهینه سازی هستند که پیدا کردن جواب کامل صحیح برای آنها امکانپذیر نمی باشد.

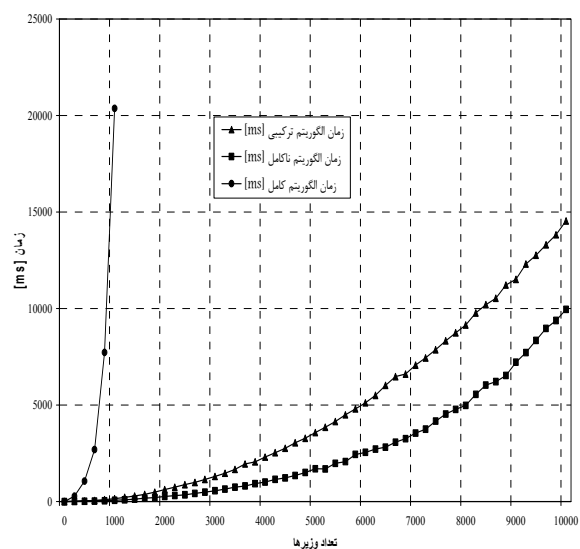
#### ۵- مراجع

- [1] E. Tsang. Foundations of Constraint Satisfaction. Academic Press, 1993.
- [2] K. Marriot, P. J. Stuckey. Programming with Constraints: An Introduction. The MIT Press, 1998.
- [3] Vipin Kumar. Algorithms for constraint satisfaction problems: A survey. AI Magazine, 13(1):32-44, 1992.
- [4] Z. Michalewicz and D. B. Fogel. How to Solve It: Modern Heuristics. Springer-Verlag, 2000.
- [5] W. Ruml. Incomplete tree search using adaptive probing. In Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-01), 2001.
- [6] Stefan Voß. Meta-heuristics: State of the art. In Alexander Nareyek, editor, Local search for planning and scheduling: revisited papers, pages 1-23. Springer-Verlag LNCS 2148, 2001.
- [7] Narendra Jussien and Olivier Lhomme. Local search with constraint propagation and conflict-based heuristics. Artificial Intelligence, 139(1):21-45, 2002.

جدول (۱) زمان لازم برای حل مسئله N-وزیر توسط الگوریتم ها

الگوریتم کامل	الگوریتم ناکامل	الگوریتم ترکیبی	تعداد وزیرها
20373 ms	70 ms	156 ms	1000
-----	270 ms	623 ms	2000
-----	562 ms	1305 ms	3000
-----	1013 ms	2307 ms	4000
-----	1696 ms	3575 ms	5000
-----	2551 ms	5115 ms	6000
-----	3553 ms	7072 ms	7000
-----	4983 ms	9149 ms	8000
-----	7220 ms	11516 ms	9000
-----	9952 ms	14543 ms	10000

همانطور که ملاحظه می شود زمان الگوریتم کامل در مقایسه با دو الگوریتم دیگر بسیار بالا است و این به دلیل کامل و دقیق بودن این نوع الگوریتم هاست که هم می توانند تمام جواب های یک مسئله را پیدا کنند و هم اینکه فضای مسئله را بصورت سیستماتیک و بترتیب جستجو می کنند. در مقایسه با الگوریتم ناکامل زمان الگوریتم ترکیبی نزدیک و متناسب با الگوریتم برخورد-کمینه [1,2] است که سریعترین الگوریتم برای حل مسئله N-وزیر می باشد. این مقایسه بصورت واضح در شکل (۱) نشان داده شده است.



شکل (۱) مقایسه سرعت الگوریتم ها برای حل مسئله N-وزیر