

# یک روش ترکیبی حاصل از الگوریتم کلونیمورچه، شبیهسازی حرارت و جستجویممنوع برای حل مسئله تخصیص دوبعدی

شیرین خضری<sup>۱</sup>، مهنوش اسدیان<sup>۲</sup>

## چکیده

الگوریتم کلونی مورچه یک سیستم چند عاملی است که از رفتار مورچهها برای حل مسائل بهینهسازی الهام گرفته است. در این مقاله مدلی از الگوریتم بهینهسازی کلونی مورچه ارائه شده است که برای بهبود عملکرد خود از یک فرایند جستجوی محلی مبتنی بر الگوریتم شبیهسازی حرارت و جستجوی ممنوع بهره گرفته است. از آنجایی که QAP یکی از مسائل NP و در عین حال یکی از جذابترین مسائل بهینهسازی ترکیبی حاضر است، برای بررسی رفتار الگوریتمهای جستجوی مکاشفهای مورد توجه قرار گرفته است. در این مقاله، الگوریتم پیشنهادی روی مسئله انتساب دوبعدی اعمال می شود. نتایج حاصل ضمن موفقیت روش پیشنهادی در حل مسئله انتساب دوبعدی، برتری قابل توجهی را در عملکرد این روش نسبت به سایر الگوریتمهای مکاشفهای نشان میدهد.

## کلمات کلیدی

بهینهسازی، کلونیمورچه، شبیهسازیحرارت، جستجویممنوع، مسئلهتخصیصدوبعدی

---

<sup>۱</sup> دانشجوی کارشناسی ارشد هوش مصنوعی، دانشگاه آزاد اسلامی واحد قزوین

آدرس پست الکترونیکی: Shirin.khezri@gmail.com

<sup>۲</sup> دانشجوی کارشناسی فناوری اطلاعات، دانشگاه علمی کاربردی واحد بیجار

آدرس پست الکترونیکی: Mehnoshasadyan@yahoo.com

## ۱. مقدمه

الگوریتم کلونی مورچه برای اولین بار توسط دوریگو و همکارانش به عنوان یک راه حل چند عامل<sup>۱</sup> برای مسائل مشکل بهینه سازی فروشنده دورگرد<sup>۲</sup>، مسئله تخصیص دوبعدی<sup>۳</sup> و ... ارائه شده است [۱].

الگوریتم کلونی مورچه الهام گرفته شده از مطالعات و مشاهدات روی کلونی مورچه هاست. این مطالعات نشان داده که مورچه ها حشراتی اجتماعی هستند که در کلونی ها زندگی می کنند و رفتار آنها بیشتر در جهت بقاء کلونی است تا در جهت بقاء یک جزء از آن. یکی از مهمترین و جالبترین رفتار مورچه ها، رفتار آنها برای یافتن غذا است و به ویژه چگونگی پیدا کردن کوتاهترین مسیر میان منابع غذایی و آشیانه. این نوع رفتار مورچه ها دارای نوعی هوشمندی توده ای است که اخیراً مورد توجه دانشمندان قرار گرفته است . مسئله تخصیص دوبعدی یک مسئله بهینه سازی ترکیبی با زمان اجرای غیرخطی است که توسط

Beckman (1957), Koopmans معرفی شد [۲]. تعداد زیادی از روشهای فرامکاشفهای نظیر جستجوی ممنوع<sup>۴</sup> [۳و۴]، شبیهساز حرارت<sup>۵</sup> [۵]، الگوریتم ژنتیک<sup>۶</sup> [۶]، الگوریتم کلونیمورچه [۷و۸] و ... برای حل مسئله تخصیص دوبعدی استفاده شده اند.

این مقاله روش جدیدی برای حل مسئله تخصیص دوبعدی ارائه میدهد که ترکیبی از کلونی مورچه با یک جستجوی محلی است که بر پایه الگوریتم شبیهسازی حرارت قرار دارد و از لیست ممنوع برای افزایش دقت استفاده شده است که با یک سری نمونههای استاندارد تست شده است. در بخش ۲، الگوریتم کلونی مورچه به طور کلی تشریح شده است، در بخش ۳ مسئله تخصیص

دوبعدی تعریف شده است و در بخش ۴ الگوریتم پیشنهادی با جزئیات ذکر شده است و سرانجام کاربرد مسئله QAP و نتایج آزمایشات آمده است که با دیگر الگوریتم ها نیز مقایسه شده است.

## ۲. بهینه سازی کلونی مورچهها

در هوشمندی توده ای عناصر رفتاری تصادفی دارند و بین آن ها هیچ نوع ارتباط مستقیمی وجود ندارد و آنها تنها بصورت غیر مستقیم و با استفاده از نشانه ها با یکدیگر در تماس هستند. مثالی در این مورد رفتار موریانه ها در لانه سازیست . این الگوریتم بر اساس کاوش طبیعی مورچههاست. مورچهها بر اساس یک مکانیزم خیلی ساده ارتباطی برای یافتن کوتاهترین مسیر بین هر دو نقطه استفاده میکنند که این مکانیزم توسط دنباله فرمونهایی که روی زمین میگذارد هدایت میشود.

بعد از مقداردهی اولیه فرمون، در هر تکرار ابتدا مورچه یک راهحل ممکن را میسازد و راهحل با اعمال یک جستجوی محلی بهبود پیدا میکند و نهایتاً فرمون به روز میشود و این پروسه تا زمان برقراری شرط توقف ادامه مییابد.

مورچه ها هنگام راه رفتن از خود ردی از ماده شیمیایی فرمون بجا می گذارند البته این ماده بزودی تبخیر می شد ولی در کوتاه مدت بعنوان رد مورچه بر سطح زمین باقی می ماند. یک رفتار پایه ای ساده در مورچه های وجود دارد ، آنها هنگام انتخاب بین دو مسیر بصورت احتمالات مسیری را انتخاب می کنند که فرمون بیشتری داشته باشد یا به عبارت دیگر مورچه های بیشتری قبلاً از آن عبور کرده باشند. در شکل ۱ نشان داده شده است که همین یک تمهید ساده چگونه منجر به پیدا کردن کوتاهترین مسیر خواهد شد.

برداشته شود و فرومون تبخیر نشود مورچه ها همان مسیر قبلی را طی خواهند کرد. ولی در حقیقت این طور نیست. تبخیر شدن فرومون و احتمال به مورچه ها امکان پیدا کردن مسیر کوتاهتر جدید را می دهند.

### ۳. تعریف مسئله

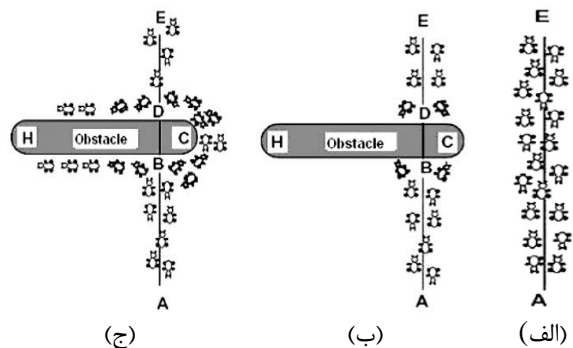
- مسئله انتساب دوبعدی میتواند به عنوان مسئله ای توصیف شود که تعداد  $n$  فعالیت را به  $n$  مکان منسوب کند به گونهای که حداقل فاصله بین محلها و همچنین جریان بین فعالیتها را می دهد. هدف جایگذاری فعالیتها در مکان هاست به گونهای که مجموع حاصلضرب بین جریانها و فواصل حداقل شود. به صورت قراردادی تعداد  $n$  فعالیت ممکن و  $n$  محل و دو ماتریس  $n \times n$  داده میشود.

$A = [a_{ij}]$ , فاصله بین موقعیت  $i, j$ ,  $B = [b_{kl}]$ , جریان بین فعالیت  $k, l$  میباشد. مسئله به صورت زیر تعریف میشود:

$$\min f(\pi) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} b_{kl} \quad (1)$$

### ۴. کاربردها

مسئله انتساب دوبعدی کاربردهای فراوانی در مدل کردن مسائلی در دنیای واقعی دارد، یک کاربرد بدیهی از آن در تئوری مکان در [۹] مطرح شد که در مدل طرح ریزی محوطه استفاده شده است. مسئله شامل طرح ریزی محل  $n$  ساختمان در یک محوطه است، بطوریکه که  $b_{kl}$  فاصله از محل  $k$  به محل  $l$  و  $a_{ij}$  حجم ترافیک بین ساختمان  $i$  و  $j$  است و هدف به حداقل رساندن فاصله پیاده رفتن هفتگی بین ساختمانهاست. بهعلاوه در مسائل مکان یابی، QAP در کاربردهایی از قبیل مسائل ترتیب و آرایش، ساخت و تولید



شکل ۱- نحوه یافتن کوتاهترین مسیر توسط کلونی مورچهها

همانطور که در شکل ۱ می بینیم مورچه های روی مسیر AE در حرکت اند (در دو جهت مخالف) اگر در مسیر مورچه ها مانعی قرار دهیم (ب) مورچه ها دو راه برای انتخاب کردن دارند. اولین مورچه از A می آید و به C می رسد، در مسیر هیچ فرومونی نمی بیند بنابراین این برای مسیر چپ و راست احتمال یکسان می دهد و بطور تصادفی و احتمالاتی مسیر ACE را انتخاب می کند. اولین مورچه ای که مورچه اول را دنبال می کند زودتر از مورچه اولی که از مسیر AHE رفته به مقصد می رسد. مورچه ها در حال برگشت و به مرور زمان یک اثر بیشتر فرومون را روی ACE حس می کنند و آنرا بطور احتمالی و تصادفی (نه حتما و قطعا) انتخاب می کنند. در نهایت مسیر ACE بعنوان مسیر کوتاهتر برگزیده می شود. در حقیقت چون طول مسیر ACE کوتاهتر است زمان رفت و برگشت از آن هم کمتر می شود و در نتیجه مورچه های بیشتری نسبت به مسیر دیگر آنرا طی خواهند کرد چون فرومون بیشتری در آن وجود دارد. اگر تصادفا اولین مورچه مسیر AHE (مسیر دورتر) را انتخاب می کرد و ردی از فرومون بر جای می گذاشت آنگاه همه مورچه ها بدنبال او حرکت می کردند و هیچ وقت کوتاهترین مسیر یافته نمی شد. بنابراین تصادف و احتمال نقش عمده ای در ACO بر عهده دارند. نکته دیگر مسئله تبخیر شدن فرومون بر جای گذاشته شده است. برفرض اگر مانع در مسیر AE

### Solution manipulation

$\pi^{k'} = T$  swaps of  $\pi^k$  based on pheromone matrix

$\pi^{k'}$  = Local search method based on Simulated Annealing ( $\pi^{k'}$ )

If  $f(\pi^{k'}) < f(\pi^k)$  Then

The best solution  $\pi^* = \pi^{k'}$

### Pheromone trail updating

Update of pheromone matrix

### Diversification

If  $n/2$  iteration have been performed without amelioration of  $\pi^*$  then Diversification

Print the best permutation and stop

Empty the tabu lists of all the ant

شکل ۲- الگوریتم شبیه سازی مورچه - ترکیب الگوریتم کلونی مورچه با الگوریتم شبیه سازی حرارت و استفاده از لیست ممنوع

- الگوریتم شبیه سازی مورچه در چهار مرحله بصورت زیر بیان میشود:

- ✓ مقداردهی اولیه
- ✓ دستکاری راه حل
- ✓ به روز رسانی فرمون
- ✓ تنوع

### ۱.۵ مقداردهی اولیه

- ابتدا به ازای هر مورچه یک راهحل اولیه تولید می شود، حال برای اینکه هر فعالیت به یک موقعیت خاص نسبت داده شود و پدیده برخورد پیش نیاید از ساختار دادهای به نام لیست ممنوع استفاده میکنیم. به این صورت که هر مورچه موقعیتهای استفاده شده را در لیستی نگهداری میکند که بهوسیله فعالیت‌های

کامپیوتر، زمانبندی، فرایند ارتباطات و ... کاربرد دارد. در زمینه طراحی ماشین نشان دادند که QAP میتواند برای طراحی ماشین تحریر بکار رود، مسئله در واقع چیدن کلیدها روی صفحه کلید است به نحوی که زمان لازم برای نوشتن متون به حداقل برسد.  $a_{ij}$  فراوانی ظاهر شدن جفت سمبولهای  $i$  و  $j$  را نشان میدهد و ماتریس فاصله  $b_{kl}$  زمان مورد نیاز برای فشردن کلیدی در مکان  $l$  بعد از فشردن کلید در مکان  $k$  را مشخص میکند و راه حل بهینه برای QAP میانگین زمان برای نوشتن یک متن را حداقل میکند. [۵]. یک کاربرد مشابه در طراحی یک ماشین برای بهبود بردهای کنترلی به منظور به حداقل رساندن خستگی چشم است [۱۰].

### ۵. الگوریتم پیشنهادی

مورچه شبیه‌سازی شده یک متد جدید برای حل QAP میباشد که ترکیبی از ACO با SA و Taboo است و جزئیات آن در الگوریتم شکل ۲ آمده است.

### Initialization

Generate  $m$  initial solutions, each one associated to one ant

For  $k=1$  to  $m$

Repeat { for each location}

Choose, with probability given by equation(2), the facility to assign From those not yet assigned.

Put the chosen facility in the tabu list of  $k^{\text{th}}$  ant

Until the **tabu** list is full

Improve  $m$  initial solutions with the local search method based on SA

Let  $\pi^*$  be the best solution

Initialization of the pheromone matrix  $F$

For  $i=1$  to  $I^{\max}$  ( $I^{\max} = n/2$ )

For all permutaions  $\pi^k$  ( $1 \leq k \leq m$ )

منسوب نشده قابل دسترسی نیست. (  $\text{taboo}_k(s)$  ) یعنی عنصر s ام در لیست تابوی مربوط به مورچه k ام. (۲)

$$P_{ij}^k(k) = \frac{\alpha \cdot \tau_{ij} + (1 - \alpha)\eta_{ij}}{\sum_{r \neq \text{taboo}_k} \alpha \cdot \tau_{ij} + (1 - \alpha)\eta_{ir}}$$

• s با احتمال ۱۰٪ به صورت زیر محاسبه میشود: (۵)

$$\frac{\tau_{r\pi_s}^k + \tau_{s\pi_r}^k}{\sum_{j \neq r} (\tau_{r\pi_s}^k + \tau_{s\pi_r}^k)}$$

## ۵-۴- جستجوی محلی

الگوریتم پیشنهادی دارای یک جستجوی محلی بر اساس الگوریتم شبیهساز حرارت است که یک روش مناسب برای حل مسائل ترکیبی است. به جای شروع دوباره بهطور تصادفی زمانی که در یک ماکزیمم محلی گیر افتادهایم، میتوانیم اجازه دهیم که جستجو چند قدم به طرف پایین بردارد تا از ماکزیمم محلی فرار کند. حلقه میانی حرارت شبیه سازی شده کاملا شبیه تپه نوردی است. به جای انتخاب بهترین حرکت یک حرکت تصادفی را انجام می دهد. اگر حرکت واقعا شرایط را بهبود بخشد، آن حرکت همیشه اجرا میشود. علاوه بر آن الگوریتم حرکت را با احتمالی کمتر از یک انجام میدهد. احتمال به صورت نمایی با "بدی" حرکت کاهش می یابد، میزان  $\Delta E$  نشان میدهد که ارزیابی بدتر شده است. پارامتر دوم T برای تعیین احتمال استفاده می شود. در مقادیر بالاتر T، حرکت بد بیشتر اجازه عمل می یابد. در حالیکه اگر T به سمت صفر برود، این حرکات بد محدود می شوند. تا زمانی که الگوریتم کم و بیش مشابه تپه نوردی عمل کند. الگوریتم، شباهت صریحی با Annealing (پردازشی که بطور آهسته مایعی را تا زمانی که یخ ببندد سرد کند) دارد و پارامتر T با دما مطابقت دارد و می توان ثابت نمود اگر دما به اندازه کافی و به آرامی پایین آورده شود، ماده به پایین ترین ساختار سطح انرژی می رسد. این امر مانند این است که بگوییم اگر T را به اندازه کافی پایین آورده شود، الگوریتم نقطه بهینه عمومی را پیدا خواهد نمود. و جزئیات آن در الگوریتم شکل ۳ آمده است.

$$\begin{cases} 1, & \text{if } j \neq \text{taboo}_k \\ 0, & \text{otherwise} \end{cases}$$

این راه حل به وسیله یک جستجوی محلی بهینه می شود. همهی فرمونهای  $\tau_{ij}$  با یک مقدار مشابه  $\tau_0$  مقداردهی اولیه می شوند که به صورت زیر محاسبه میشود:

$$\tau_0 = \frac{1}{10 \times (f(\pi^*))} \quad (۳)$$

$\pi^*$  بهترین راه حلی است که تاکنون پیدا شده است و  $\eta_{ij}$  یک فرمون مطلوب است.

## ۲.۵ دستکاری راه حل

دستکاری راه حل مسئله دو استراتژی دارد:

- اصلاح دنباله فرمونی
- جستجوی محلی

### ۵-۲-۱- اصلاح دنباله فرمونی

این عمل روی هر راه حل مورچه اعمال میشود و به صورت  $\pi^k$  نشان میدهم. تعداد T مبادله در این مرحله اعمال می شود. مبادله به معنی جابجایی دو فعالیت از یک جایگشت است. ابتدا فعالیت r به صورت تصادفی انتخاب می شود و سپس فعالیت s طبق یکی از دو روش زیر انتخاب میشود.

• s با احتمال ۹۰٪ به صورت زیر محاسبه میشود: (۴)

$$\max (\tau_{r\pi_s}^k + \tau_{s\pi_r}^k)$$

(۷)

$$\tau_{i\pi_i^*} = \tau_{i\pi_i^*} + \frac{\alpha}{f(\frac{\pi^* + \pi^{*-1} + \pi^{*-2}}{3})}$$

$\pi^*$  بهترین راه حلی است که تاکنون پیدا شده است.

#### ۴.۵. نوع یا گوناگونی

این مکانیزم زمانی اعمال می شود که تا تکرار  $n/2$  بهبودی حاصل نشود. مکانیزم گوناگونی این امکان را فراهم می کند که الگوریتم روی راه حلهایی با ساختار مختلف کار کند. ابتدا قبل از اینکه مکانیزم گوناگونی شروع شود بهترین سه راه حل یافت شده نگهداری می-شوند، سپس همه اطلاعات نگهداری شده دنباله فرمونی پاک میشود و ماتریس فرمونها دوباره مقداردهی می شود. سپس یک راه حل جاری جدید برای همه مورچه ها به صورت تصادفی تولید میشود، اما برای سه مورچه همان سه راه حل نگهداری شده استفاده میشود.

#### ۶. نتایج تجربی

برای ارزیابی کارایی الگوریتم پیشنهادی، نتایج شبیه سازی با تعدادی از الگوریتمهای متاهیورستیک مقایسه شده است. در این مقایسهها، نمونهها که از مجموعه داده QAPLIB [۱۱]، گرفته شدهاند، به دو دسته منظم و نامنظم تقسیم شدهاند. این تقسیم بندی با استفاده از یک روش ساده ایستا به اسم Flow-Dominance انجام شده است [۱۲]. بطوریکه نمونههای منظم دارای Flow-Dominance کمتر از ۱.۲ در مقابل نمونههای نامنظم هستند. همچنین نمونههای نامنظم را گاهها نمونههای ساخت یافته مینامند. حل مسائل این بخش در شرایط یکسان انجام شدهاند. به اینصورت که جمعیت مورچهها ۲۰، تعداد تکرارها ۱۰۰ مرتبه، تعداد جابجاییها ۱۰ و ضریب تبخیر ۰.۱ مقدار دهی میشوند. این پارامترها برای ۱۰ مرتبه

#### Step1

$c=0$ ;

Determine the initial permutation ( $\pi^k$ ), initial Temperature ( $T_0$ ), last temperature ( $T_s$ ), a cooling proportion ( $\tau$ );

#### Step2

Create a new permutation ( $\pi_j$ ) from current Permutation ( $\pi_i$ ) by using Selected Neighborhood mechanism;

$$\Delta = f(\pi_j) - f(\pi_i);$$

If  $\Delta < 0$  Then

$$\pi_i := \pi_j;$$

Else if random  $(0.1) < \exp(-\Delta/T)$  Then

$$\pi_i := \pi_j;$$

#### Step3

$$T_{c+1} = T_c / (1 + T_c);$$

$$c := c + 1;$$

If  $c > k$  Then Stop;

Else Go to Step

شکل ۳- الگوریتم جستجوی محلی مبتنی بر SA

#### ۳.۵ به روز رسانی دنباله فرمونی

این کار در دو فاز انجام میشود، ابتدا دنباله فرمونی با شبیهسازی فرایند تبخیر به روز میشود که با کاهش مقدار فرمون به صورت زیر انجام میشود:

$$\tau_{ij} = (1 - \alpha) \cdot \tau_{ij}, 1 \leq i, j \leq n \text{ and } 0 < \alpha < 1 \quad (۶)$$

اگر مقدار  $\alpha$  صفر باشد فرمون به مدت طولانی باقی می ماند و اگر مقدار  $\alpha$  به یک نزدیک باشد میزان تبخیر بالا خواهد بود.

فاز بعدی این است که دنباله فرمونی تقویت شود. این استراتژی شامل تابعی از ۳ تا از بهترین راه حلهای پیدا شده برای به روز رسانی دنباله فرمونی است که به صورت معادله ۷ محاسبه میشود:

اجرای مستقل برنامه محاسبه شده و متوسط نتایج در جدولهای ۱ و ۲ آمده است.

جدول ۱ نتایج نمونههای منظم را نشان میدهد که الگوریتم پیشنهادی با دیگر الگوریتم ها مقایسه شده است که دارای بهترین نتیجه از بین دیگر الگوریتمها میباشد. نمونههای نامنظم در جدول ۲ نشان داده شده- اند.

## ۷. نتیجه گیری

در این مقاله الگوریتم قدرتمندی برای حل مسئلهی

- QAP پیشنهاد شده است که بر پایهی کلونی مورچه هاست. فرایند جستجوی هریک از مورچهها با یک

- الگوریتم جستجوی محلی SA و لیست تابو میباشد و عمل یادگیری را نیز انجام میدهد. مورچه شبیه سازی شده یک استراتژی جدید برای یادگیری دنباله فرمونی دارد به این صورت که ۳ تا از بهترین راه حلها یافت شده را برای بروز رسانی دنباله فرمونی نگهداری می کند. نتایج مقایسات بهترین الگوریتمهای هیورستیک با این الگوریتم در جداول ۱ و ۲ آمده اند. نتایج نشان می دهد که کارایی این الگوریتم در مقایسه با دیگر الگوریتمها بهتر بوده و این به دلیل ترکیب کلونی مورچهها با الگوریتم جستجوی محلی شبیه سازی حرارت و استفاده از لیست تابو بوده است.

جدول ۱- نتایج بدست آمده روی نمونههای منظم به بطور متوسط بعد از ۱۰ اجرا

Problem	Best known value	TS (gap)[4]	SA (gap)[5]	GA (gap)[6]	ACO (gap)[8]	Value of Proposed Ant	Proposed Ant(gap)
nug20	2570	0.911	0.071	0	0	2570	0
nug30	6124	0.872	0.121	0.007	0	6124	0
sko42	15812	1.116	0.114	0.003	0	15816	0.0002
sko49	23386	0.978	0.133	0.040	0.027	23406	0.0008
sko56	34458	1.082	0.110	0.060	0.001	34558	0.002
sko64	48498	0.861	0.095	0.092	0.041	48578	0.0016
sko90	115534	0.748	0.227	0.196	0.136	115534	0
tai20a	703482	0.246	0.716	0.268	0.255	705622	0.003
tai30a	1818146	0.286	0.907	0.439	0.454	1844186	0.0143
tai35a	2422002	0.355	1.345	0.698	0.440	2462768	0.0168
tai60a	7208572	0.831	1.395	1.159	0.964	7359954	0.021
tai80a	13557864	0.467	0.995	0.796	0.362	13768756	0.0155
wi150	48816	0.504	0.061	0.032	0.088	48856	0.0008

جدول ۲- نتایج بدست آمده روی نمونههای نامنظم به طور متوسط بعد از ۱۰ اجرا

Problem	Best known value	TS(gap)	SA(gap)	GA(gap)	ACO(gap)	Value of Proposed Ant	Proposed Ant(gap)
bur26a	5426670	0.0004	0.1411	0.0120	0	5426670	0
bur26d	3821225	0.0015	0.0056	0.0002	0	3821225	0
bur26e	5386879	0	0.1238	0	0	5386879	0
bur26f	3782044	0.0007	0.0.1579	0	0	3782044	0
bur26g	10117172	0.0003	0.1688	0	0	10117172	0

<b>bur26h</b>	7098658	0.0027	0.1268	0.0003	0	7098658	0
<b>chr25a</b>	3796	6.9652	12.4973	2.6923	1.2448	3796	0
<b>els19</b>	17212548	0	18.5385	0	0.7764	17212548	0
<b>kra30a</b>	88900	0.4702	1.4657	0.1338	0	88900	0
<b>kra30b</b>	91420	0.0591	0.1974	0.0536	0	91420	0
<b>tai50b</b>	458821517	0.2943	0.8107	0.2142	0.2475	458821517	0
<b>tai60b</b>	608215054	0.3904	2.1373	0.2905	0.2258	608215054	0
<b>tai80b</b>	818415043	1.4354	1.4386	0.8286	0.8214	818415043	0

✓ اختلاف بين بهترين جواب و جواب بدست آمده از روش پيشنهادهی به صورت زیر محاسبه میشود:

(A)

$$\text{Gap} = (\text{Best known value} - \text{Solution value}) / \text{Best known value}$$

Transportation Research 6, pp. 59–68, 1972.

- [10] R. E. Burkard, J. Offermann, *Entwurf von Schreibmaschinentastaturen mittel Quadratischer Zuordnungs Problem*, Z. Operations Research, 21, pp. 121–132, 1997.
- [11] R. E. Burkard, S. Karisch, F. Rendl, *Qaplib: a Quadratic Assignment Problem Library*, European Journal of Operational Research 55, pp. 115-119, 1991.
- [12] U. Venkatadri, *A Note on the Flow Dominance Measure*, IENG 4445-Facilities Design, pp. 871-879, 2006.

۸. مراجع

- [1] M. Dorigo, *Optimization, Learning and Natural Algorithms*, Ph.D. Thesis, Politecnico di Milano, Italy, 1992.
- [2] T. C. Koopmans, M. J. Beckmann, *Assignment Problems and the Location of Economics Activities*, Econometrica 25, pp. 53–76, 1957.
- [3] J. Skorin-Kapov, *Tabu Search applied to the Quadratic Assignment Problem*, ORSA. Journal on Computing, 2, pp. 33–45, 1990.
- [4] E. D. Taillard, *Robust Taboo Search for the Quadratic Assignment Problem*, Parallel Computing, 17, pp. 443–455, 1991.
- [5] D.T. Connolly, *An Improved Annealing Scheme for the QAP*, European Journal of Operational Research 46, pp. 93–100, 1990.
- [6] L. Randy, S. Ellen Haupt, *Practical Genetic Algorithms*, 2<sup>nd</sup> Edition, John Wiley & Sons, Inc, pp. 321-331, 2004
- [7] T. Stutzle, M. Dorigo, *ACO algorithms for the Quadratic Assignment Problem*, *New Ideas in Optimization*, McGraw-Hill, vol. 23, pp. 123-129, 1999.
- [8] N. Cetin Demirel, D. Toksari, *Optimization of the Quadratic Assignment Problem using an Ant Colony Algorithm*, Applied Mathematics and Computation 183, pp. 427-435, 2006.
- [9] J. W. Dickey, J. W. Hopkins, *Campus building arrangement using TOPAZ*,

زیر نویسهها

- 
- <sup>1</sup> Multi Agent  
<sup>2</sup> Travelling Salesman Problem  
<sup>3</sup> Quadratic Assignment Problem (QAP)  
<sup>4</sup> Taboo Search  
<sup>5</sup> Simulated Annealing  
<sup>6</sup> Genetic Algorithm  
<sup>7</sup> Ant Colony