



## Jump of Circles: A New Way to Solve the Engineering Optimization Problems

N. Haji Aghajanpour, M. R. Ghasemi\*, H. Ghohani Arab

Department of civil engineering, University of Sistan and Baluchestan, Zahedan, Iran

**ABSTRACT:** In this paper, a new meta-heuristic optimization method called the Jump of Circles Optimization Method is introduced. In any optimization problem, an answer zone is defined in which the optimization algorithms search the space to find the optimal answer. The method presented in this paper uses two important pillars in searching the answer zone. The first pillar is to use geometric principles. The Jump of circles uses the circle with decreasing radius. The second pillar is to use the meta-heuristic application. In meta-heuristic algorithms, the search points distribute randomly and jump in the answer zone. In the proposed method, the center of the searching circle jumps and sits on the optimal point of each step. The proposed algorithm solves the optimization problem in two phases. The first phase is optimal area exploration and the second phase is exploiting the exploration. Finally, the most optimal point that will be obtained from the two phases, is the optimal answer to the problem.

### Review History:

Received: Oct. 15, 2019  
Revised: May, 30, 2019  
Accepted: Jun. 16, 2019  
Available Online: Jul. 14, 2019

### Keywords:

Jump Of Circles Optimization Method  
Meta-Heuristic Method  
Two Phases Method  
Optimization  
Engineering Benchmark Problems.

### 1- Introduction

Optimization is a very applicable tool used in almost all sciences. With this highly powerful tool, any problem that can be shown mathematically can be solved. High applicability and problem-solving capability have increased the importance of optimization among researchers in recent years. The Genetic Algorithm (GA) is a highly applicable algorithm used in many optimization problems [1]. Such other researches are the Particle Swarm Optimization method (PSO) [2], the Ant Colony Optimization method (ACO) [3], and Gravitational Search Algorithm (GSA) [4].

There are two different phases in the “Jump of Circles Optimization Method (JCOM)” presented in this paper: in phase 1, an effort is made to identify a part in the answer zone where there is the most optimal answer and then in phase 2, use is made of that answer to obtain the problem’s optimal answer. The two mentioned phases somehow overlap one another to find the better answers, that is, at the beginning of phase 2, the radius of the circle is so chosen that it may exceed that of the final limit in phase 1. To investigate different parts of the answer zone, use has been made, in the proposed method, of random numbers to select the study points. It is worth noting that the random points, generated by the algorithm in both phases, face circles that get smaller and smaller.

### 2- Methodology

This section explains the “Jump of Circles Optimization Method”. It is worth mentioning that the center of the circle is displaced in each step called the jump of circles. Since many of the problems are constrained, use has been made of an evaluator function as follows to evaluate points [5]:

$$EF = GF (1 \pm r_p \cdot CV^2) \quad (1)$$

$$r_p = 4 \left( 1.6^{\left[ \frac{i}{2} \right]} \right)$$

Where  $EF$  is the Evaluator Function,  $GF$  is the Goal Function,  $CV$  is the Constraint Violation and  $r_p$  is the penalty coefficient of the constraint violation in step  $i$  (it is a very useful factor used to find accurate answers). This function is based on statistical studies,  $[\ ]$  shows the integer part, and the plus and minus signs are used for the minimization and maximization, respectively. The proposed method performs the optimization in two phases as follows.

\*Corresponding author’s email: mrghasemi@eng.usb.ac.ir



2.1. Phase 1: Exploring The Optimal Area

In the first step, a point is selected in the answer zone quite randomly and then some random search points are created around it with a limited distance  $r$  which is the radius of the hypothetical circle reduced in each step according to the following linear relationship:

$$r_0 = 0.1(x_{max}^d - x_{min}^d) \tag{2}$$

$$r_i = r_0 - r_0 \times \frac{i - 1}{i_{max1}} \tag{3}$$

$$x_{i_{new}}^d = x_{i-1}^d + a.b.r_i \tag{4}$$

Where,  $x_{max}$  and  $x_{min}$  are the maximum and minimum values extractable for the desired variable, respectively,  $r_0$  is the initial radius of the circle,  $r_i$  is the radius of the circle in step  $i$ ,  $i_{max1}$  is the maximum number of steps in phase 1,  $x_i^d$  is the position of the random point in step  $i$  and in direction  $d$ ,  $x_{i_{new}}^d$  is the position of the new search point in step  $i$  and in direction  $d$ ,  $a$  is a random sign and  $b$  is a random number between 0 and 1. Next, situations of the search points are evaluated by the evaluator function and the three best ones are selected; in minimization problems (truss weight), the best situation is when  $EF$  has the least answer. The “jump of circles” algorithm considers two hypothetical lines between the best and two other points and then generates two hypothetical points on each line so that they may lie on both sides of the best point:

$$\Delta X = x_0 \cdot \left( \frac{i_{max1}}{i - 1} \right) \cdot \left( \frac{x_{max} - x_{min}}{2 \times i_{max1}} \right) \tag{5}$$

$$x_{i_{new}}^d = x_{i_{best}}^d \pm \Delta X \tag{6}$$

Here,  $x_0$  is a random number between 0 and  $\pi$ , the expression in parentheses in the right side of Eq. (5) is to fit the value of  $\Delta X$  to the problem dimensions (it is worth mentioning that the earlier searches need more random values for the algorithm’s better answers, but the more the solution continues, the less will be this need; this has been considered in the solution through the expression in parentheses in the left side of Eq. (5)), and  $x_{i_{best}}^d$  is the situation of the best point in step  $i$  and in direction  $d$ . The situations of the four new points are estimated by the evaluator function; if there is any improvement in the best point, it will be considered as the step’s best point. To start the next step, the center of the circle is positioned on the best point of the last step instead of on the random point of the first step; this will continue until phase 1 stops.

2.2. Phase 2: Exploiting the exploration

The best point of the last step in phase 1 is considered as the center of the circle at the beginning of phase 2. For the circle concept to enter the solution, building relations in this phase of the problem is done circularly in two consecutive dimensions; in other words, the relationships in the even dimension depend on those of the former odd dimension. In phase 2, some search points are created inside the hypothetical circle and some on its periphery as follows:

$$x_{i_{new}}^{d=2m-1} = x_{i_{center}}^{d=2m-1} + r_i \times \cos(\alpha) \tag{7}$$

$$x_{i_{new}}^{d=2m} = x_{i_{center}}^{d=2m} + r_i \times \sin(\alpha) \tag{8}$$

Where  $2m - 1$  shows the odd direction and  $2m$  represents the even one,  $\alpha$  is the angle of the desired point to be chosen from the  $[0-2\pi]$  interval divided equally among the points on the periphery (we suggest 4-6 points on the periphery where speed is important and 24-30 where accuracy is vital), and  $r$  is the radius of the circle reduced linearly in each step as follows:

$$r_0 = 0.03(x_{max}^d - x_{min}^d) \tag{9}$$

$$r_i = r_0 - r_0 \times \frac{i - 1}{i_{max2}} \tag{10}$$

Where,  $x_{max}$  and  $x_{min}$  are, respectively, the maximum and minimum values extractable for the desired variable,  $r_0$  is the initial radius of the circle in phase 2,  $r_i$  is the radius of the circle in step  $i$  of phase 2, and  $i_{max2}$  is the maximum number of steps in phase 2. It is worth noting that  $r_i$  reaches zero at the end of phase 1, but it begins phase 2 with about 1/3 of the initial value in phase 1; this causes the overlapping of the two phases and helps to improve the answers. In the “jump of circles” algorithm, the number of the search points of the circle is a multiple of 4 created by the following relations in such a way that their number is equal in each quadrant.

$$x_{i_{new1}}^{d=2m-1} = x_{i_{center}}^{d=2m-1} + a4 \times (a1 \times r), x_{i_{new1}}^{d=2m} \tag{11}$$

$$= x_{i_{center}}^{d=2m} + a4 \times \sqrt{r^2 - (a1 \times r)^2}$$

$$x_{i_{new2}}^{d=2m-1} = x_{i_{center}}^{d=2m-1} + a3 \times (a2 \times r), x_{i_{new2}}^{d=2m} \tag{12}$$

$$= x_{i_{center}}^{d=2m} - a3 \times \sqrt{r^2 - (a2 \times r)^2}$$

$$x_{i_{new3}}^{d=2m-1} = x_{i_{center}}^{d=2m-1} - a2 \times (a3 \times r), x_{i_{new3}}^{d=2m} \tag{13}$$

$$= x_{i_{center}}^{d=2m} + a2 \times \sqrt{r^2 - (a3 \times r)^2}$$

$$x_{i_{new4}}^{d=2m-1} = x_{i_{center}}^{d=2m-1} - a1 \times (a4 \times r), x_{i_{new4}}^{d=2m} \tag{14}$$

$$= x_{i_{center}}^{d=2m} - a1 \times \sqrt{r^2 - (a4 \times r)^2}$$

$a1$ ,  $a2$ ,  $a3$  and  $a4$  are four random numbers in the  $[0,1]$  interval and the value of the evaluator function is calculated at the search points created on the periphery of the circle and inside it; this information is then used to create the other four search points. To create the new points, we first specify half of the set-close points on the circle periphery whose sum of the evaluator function is the best value; in maximization problems, it is the maximum value and vice versa. Next, the exit direction from the circle is considered toward this half and four points with the best situations are selected from among all the search points created inside and on the circular periphery. Since these points are in the direction of the circle exit or entry, they will yield a new search point.

$$\Delta x = x_0 \cdot \left( \frac{i_{max2}}{i-1} \right) \cdot \left( \frac{x_{max} - x_{min}}{2 \times i_{max2}} \right) \quad (15)$$

$$\begin{cases} x_{i_{new}}^d = x_{i_{point}}^d + \Delta x \cdot dir & \text{If located in output direction} \\ x_{i_{new}}^d = x_{i_{center}}^d + \Delta x \cdot dir & \text{If located in input direction} \end{cases} \quad (16)$$

Where,  $i_{max2}$  is the maximum number of iterations and  $i$  is the counter of steps in phase 2. Direction is shown by  $dir$  which is actually a positive or negative number recognized as explained earlier. Situations of the new search points are evaluated by the evaluator function and one point with the best value (from among these 4 and those taken from the circle) is considered as the best point of this step; the circle

then jumps to this point for the next step and the process will continue until phase 2 stops; the best point of the last step will be the problem's optimal answer.

### 3- Conclusion

This paper introduced the "Jump of Circles Optimization Method (JCOM)" which is a meta-heuristic method combined with geometric constraints. The proposed algorithm consists of two phases; phase 1 explores the optimal zone and phase 2 exploits the exploration. Each phase is responsible for one important optimization principle, they overlap to yield better answers, and meta-heuristic applications ease taking points from different parts of the search space.

### References

- [1] Holland JH. Genetic algorithms. Scientific American, 267 (1992) 66-72.
- [2] Kennedy J, Eberhart R. Particle swarm optimization. Proceedings of the 1995 IEEE international conference on neural networks, (1995) 1942-1948.
- [3] Dorigo M, Birattari M, Stutzle T. Ant colony optimization. IEEE Computational Intelligence, 1(4) (2006) 28-39.
- [4] Rashedi E, Nezamabadi Pour H, Saryazdi S. GSA: a gravitational search algorithm. Information Science, 179(13) (2009) 2232-2248.
- [5] Naderi, A., Sohrabi, M.R., Ghasemi, M.R., Dizangian B., Total and Partial Updating Technique: A Swift Approach for Cross-Section and Geometry Optimization of Truss Structures. KSCE J Civ. Eng., 24 (2020) 1219-1227.

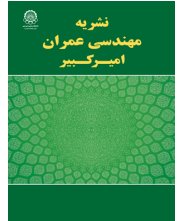
#### HOW TO CITE THIS ARTICLE

N. Haji Aghajanpour, M. R. Ghasemi, H. Ghohani Arab., Jump of Circles: A New Way to Solve the Engineering Optimization Problems. Amirkabir J. Civil Eng., 53 (5) (2021) 435-438

DOI: [10.22060/ceej.2020.17227.6496](https://doi.org/10.22060/ceej.2020.17227.6496)







## پرش دایره ها: روشی نوین برای حل مسائل بهینه‌سازی مهندسی

محمدرضا قاسمی<sup>\*</sup>، نادر حاجی آقاچان‌پور، حامد قوهانی عرب

دانشکده مهندسی شهید نیکبخت، دانشگاه سیستان و بلوچستان، زاهدان، ایران

### تاریخچه داوری:

دریافت: ۱۳۹۸/۰۷/۲۳

بازنگری: ۱۳۹۹/۰۳/۰۹

پذیرش: ۱۳۹۹/۰۳/۲۶

ارائه آنلاین: ۱۳۹۹/۰۴/۲۳

### کلمات کلیدی:

روش بهینه‌سازی پرش دایره ها

روش های فرااکتشافی

روش دو فازی

بهینه سازی

مسائل محک سازه ای

**خلاصه:** در این مقاله روش بهینه‌سازی پرش دایره ها که یک روش فرااکتشافی می باشد ارائه می گردد. در هر مسأله بهینه‌سازی یک فضای پاسخ تعریف می شود که الگوریتم‌های بهینه‌سازی با جستجو در آن فضا، پاسخ بهینه را می یابند. روش پیشنهاد شده در این مقاله از دو رکن مهم در جستجوی فضای پاسخ بهره می گیرد. رکن اول استفاده از اصول هندسه می باشد. در روش پرش دایره ها، از شکل دایره که در طول حل اندازه شعاع آن کاهش می یابد، بهره گیری شده است. رکن دوم کاربرد فرا-اکتشافی است. آنچه در الگوریتم‌های فرااکتشافی مشاهده می شود، پخش شدن تصادفی نقاط مورد بررسی در فضای پاسخ است. در روش پرش دایره‌ها که در این مقاله ارائه می شود، مرکز دایره مورد جستجو به بهینه ترین نقطه هر گام پرش می کند. الگوریتم ارائه شده شامل دو فاز می باشد. فاز اول اکتشاف بهینه ترین محدوده و فاز دوم بهره برداری از اکتشاف است. در پایان بهینه ترین نقطه گام آخر فاز دوم، پاسخ بهینه مسأله خواهد بود. در این مقاله با توجه به اینکه روش پیشنهادی با تأکید بر مسائل مهندسی ارائه می شود، سه مسأله محک خرپا حل شده است. همچنین برای نشان دادن توانایی روش پیشنهادی مسأله پیچیده کین نیز با آن حل گردیده است. پاسخ‌های این مسأله ها با تعدادی از روش های مرسوم مقایسه شده و در جداول جداگانه ارائه می گردد. در نتایج روش پرش دایره ها، بهبود قابل ملاحظه ای مشاهده می شود

### ۱- مقدمه

های بعد تکامل می یابند. در واقع در الگوریتم ژنتیک عملگر جهش، مسئولیت پرش نقاط جستجو به قسمتهای مختلف فضای پاسخ را بر عهده دارد. از دیگر پژوهشها می توان به روش بهینه سازی برپایه آموزش و یادگیری [۲] جستجوی هماهنگی [۳] جستجوی ممنوع [۴] بهینه ساز جستجوی گروهی [۵] استراتژی تکاملی [۶] برنامه-ریزی ژنتیک [۷] بهینه ساز جغرافیای زیستی [۸] و آموزش افزایشی برپایه احتمال [۹] نام برد.

یکی دیگر از الگوریتمهای مشهور، روش بهینه سازی اجتماع ذرات است [۱۰] در این روش تعدادی از ذرات در یک فضای جستجو برای یافتن بهترین جواب با الگوی خاصی حرکت میکنند. روش بهینه سازی اجتماع مورچگان [۱۱] بهینه ساز جفتگیری پرنده

بهینه سازی یکی از پرکاربردترین ابزارهایی است که تقریباً در تمام علوم از آن بهره گیری شده است. به کمک این ابزار بسیار توانا، هر مسأله‌ای که بتواند به شکل ریاضی درآید قابل حل است. در سالهای اخیر کاربرد زیاد و توانایی حل مسأله، اهمیت بهینه سازی را در بین پژوهشگران افزایش داده است. از جمله پژوهشهایی که در زمینه بهینه سازی انجام شده و شهرت و کاربرد فراوانی پیدا کرده است، الگوریتم ژنتیک میباشد که بر اساس نظریه تکامل داروین شکل گرفته است [۱] الگوریتم ژنتیک از سری الگوریتمهای تکاملی است که جمعیت اولیه‌ای به صورت تصادفی ایجاد می شود و در نسل-

<sup>\*</sup> نویسنده عهده‌دار مکاتبات: mrghasemi@eng.usb.ac.ir



شود. شایان ذکر است در هر گام مرکز دایره جابجا میشود که در این مقاله از آن به عنوان پرش دایره یاد شده است. با توجه به این که این مقاله با تأکید بر مسائل مهندسی عمران نوشته شده است و بسیاری از مسائل مهندسی، مقید می باشند، از مفهومی به نام تابع ارزیاب برای بررسی نقاط بهره گیری شده است:

$$EF = GF(1 \pm r_p \cdot CV^2) \quad (1)$$

$$r_p = 4 \left( 1.6^{\left[ \frac{i}{2} \right]} \right)$$

در رابطه بالا،  $EF$  ابع ارزیاب،  $GF$  ابع هدف،  $CV$  میزان تخطی قیود و  $r_p$  ریب جریمه تخطی قیود در گام  $i$ ام میباشد.  $r_p$  ریبی بسیار کاربردی برای رسیدن به پاسخ دقیق می باشد. رابطه (۱) عملکرد بسیار خوبی در همگرایی به پاسخ دقیق دارد و به همین سبب مورد استقبال پژوهشگران واقع شده است [۲۰]. ضرایب رابطه ارائه شده بر اساس بررسیهای آماری می باشد که همگرایی بهتری را در مسائل مورد بررسی نتیجه میدهد. علامت  $+$  جزء صحیح را نشان میدهد. در این رابطه علامت  $+$  برای کمینه یابی و علامت  $-$  برای بیشینه یابی کاربرد دارد. روش پیشنهادی بهینه سازی را در دو فاز به شرحی که در ادامه میآید انجام میدهد.

#### ۱-۲- فاز اول: اکتشاف بهینه ترین محدوده

روابطی که در این بخش ارائه می شود برای یک بعد از مسأله (بعد  $d$ ) می باشد که به راحتی برای تک تک ابعاد مسأله نیز قابل استفاده است. در گام اول، یک نقطه از منطقه پاسخ به صورت کاملاً تصادفی انتخاب می شود، سپس تعدادی نقطه جستجوی تصادفی با محدودیت فاصله  $r$  طرف آن ایجاد می گردد.  $r$  مان شعاع دایره فرضی می باشد که در هر گام با رابطه خطی زیر کاهش می یابد:

$$r_0 = 0.1(x_{max}^d - x_{min}^d) \quad (2)$$

$$r_i = r_0 - r_0 \times \frac{i-1}{i_{max1}} \quad (3)$$

$$x_{i_{new}}^d = x_{i-1}^d + a \cdot b \cdot r_i \quad (4)$$

[۱۲] الگوریتم بهینه سازی مگس میوه [۱۳] دسته جالبکها [۱۴] شبیهسازی سختشونده [۱۵] الگوریتم جستجوی محلی گرانش [۱۶] انفجار بزرگ-بحران بزرگ [۱۷] بهینه ساز نیروی مرکزی [۱۸] و الگوریتم جستجوی گرانش [۱۹] از جمله دیگر فعالیتهایی است که در زمینه بهینه سازی صورت گرفته است.

مطالعه روشهای بهینه سازی ارائه شده نشان میدهد برای آن که روش بهینه سازی پاسخهای قابل قبولی داشته باشد، اصولی باید رعایت شود. نخست این که در فرایند جستجو در منطقه پاسخ ابتدا اکتشاف بهترین محدوده صورت گیرد و سپس از نتایج این اکتشاف در یافتن پاسخ بهینه بهره برداری شود. دوم این که برای جستجوی بهتر بایستی نقاط از قسمت های مختلف منطقه پاسخ مورد بررسی قرار گیرد. در روش پرش دایره ها که در این مقاله ارائه میشود، دو فاز مختلف در نظر گرفته شده است. در فاز اول که اکتشاف بهینه ترین محدوده نام دارد تلاش می شود محدوده ای از منطقه پاسخ که بهینه ترین پاسخها در آن قسمت موجود است شناسایی شود. سپس در فاز دوم که بهره برداری از اکتشاف نام دارد، از بهینه ترین پاسخهای فاز اول برای رسیدن به پاسخ بهینه مسأله بهره گیری می شود. در روش پیشنهادی برای رسیدن به پاسخهای بهتر، نوعی همپوشانی بین دو فاز در نظر گرفته شده است، به این گونه که شعاع دایره در شروع فاز دوم به گونهای انتخاب میشود که از محدوده نهایی فاز اول فراتر رود. برای رعایت اصل دوم که همان بررسی قسمت های مختلف منطقه پاسخ می باشد، در روش ارائه شده از اعداد تصادفی برای انتخاب نقاط مورد بررسی بهره گیری شده است. البته نقاط تصادفی تولید شده توسط الگوریتم در هر دو فاز با محدودیت دایره کوچک شونده مواجه هستند. آنچه در ادامه میآید توضیحی بر روش بهینه سازی پرش دایره ها می باشد و در پایان مسائل محک خرپا و مسأله محک کین به کمک این روش حل می شوند و با تعدادی از نتایج موجود در ادبیات مقایسه می گردند. بهبود پاسخهای به دست آمده از روش پیشنهادی در جداول ارائه شده به روشنی نمایان است.

#### ۲- روش بهینه سازی پرش دایره ها

در این بخش الگوریتم بهینه سازی پرش دایرهها توضیح داده می شود. این روش در واقع روشی فرااکتشافی با محدودیت هندسی است. در این روش تلاش شده محدودیتهای هندسی با کمک دایره لحاظ

جستجو نیاز به مقادیر تصادفی بیشتری می باشد اما هر چه حل پیشتر رود، این نیاز کمتر می شود. این مورد به کمک عبارت داخل پرانتز سمت چپ رابطه (۵) وارد حل شده است.  $x_{i\ best}^d$  موقعیت بهترین نقطه در گام  $i$  و جهت  $d$  می باشد.

وضعیت چهار نقطه جدید به کمک تابع ارزیاب، بررسی می شود. چنانچه در هر یک از آنها بهبودی نسبت به بهترین نقطه مشاهده شد، آن نقطه به عنوان بهترین نقطه گام لحاظ می شود. برای شروع گام بعدی، موقعیت مرکز دایره را به جای نقطه تصادفی گام اول روی بهترین نقطه آخرین گام قرار می دهیم. این عملیات تا رسیدن به شرط توقف فاز اول ادامه می یابد.

### ۲-۲- فاز دوم: بهره برداری از اکتشاف

بهترین نقطه از گام آخر فاز اول، به عنوان مرکز دایره در شروع فاز دوم در نظر گرفته می شود. برای آنکه مفهوم دایره به معنای واقعی آن وارد حل شود، در این فاز رابطه سازی ها در دو بعد متوالی از مسأله به صورت دایروی انجام می شوند. به عبارت دیگر رابطه ها در بعد زوج به بعد فرد پیشین خود وابسته است. بنابراین روابطی که در این بخش ارائه می شود برای دو بعد متوالی از مسأله (بعد فرد  $d$  و بعد زوج  $d + 1$ ) می باشد که به راحتی برای سایر ابعاد مسأله نیز قابل استفاده است. در فاز دوم تعدادی نقطه جستجو روی محیط دایره فرضی و تعدادی درون دایره ایجاد می شود. نقاط روی محیط به صورت زیر ایجاد می شوند:

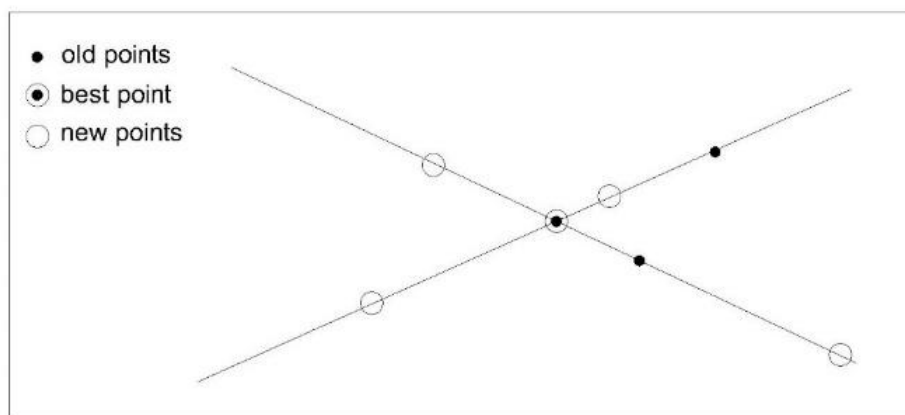
در رابطه های ارائه شده  $x_{\ max}^d$ ،  $x_{\ min}^d$  ترتیب حداکثر و حداقل مقادیر قابل برداشت برای متغیر  $d$ ،  $r_0$  عاع اولیه دایره،  $r_i$  عاع دایره در گام  $i$  و  $i_{\ max\ 1}$  حداکثر تعداد گام در فاز اول است.  $x_{i-1}^d$  ر گام اول موقعیت نقطه تصادفی و در گامهای بعد موقعیت بهترین نقطه گام قبل در جهت (بعد)  $d$   $x_{i\ new}^d$  موقعیت نقطه جستجوی جدید در گام  $i$  و در بعد  $a$   $d$  علامت تصادفی و  $b$   $k$  عدد تصادفی بین ۰ و ۱ می باشند.

در ادامه به کمک تابع ارزیاب وضعیت نقاط جستجو ارزیابی می شود و سه نقطه که بهترین وضعیت را دارا هستند برداشت می شود. منظور از بهترین وضعیت در مسائل کمینه یابی (وزن خرپا) حالتی است که تابع ارزیاب  $EF$  دارای کمترین پاسخ باشد. الگوریتم پرش دایره ها بین بهترین نقطه و دو نقطه دیگر دو خط فرضی را در نظر می گیرد، سپس روی هر کدام از خطوط دو نقطه فرضی را به گونهای ایجاد می کند که در دو طرف بهترین نقطه قرار گیرند (شکل ۱):

$$\Delta x = x_0 \cdot \left( \frac{i_{\ max\ 1}}{i - 1} \right) \cdot \left( \frac{x_{\ max} - x_{\ min}}{2 \times i_{\ max\ 1}} \right) \quad (5)$$

$$x_{i\ new}^d = x_{i\ best}^d \pm \Delta x \quad (6)$$

$x_0$  ک عدد تصادفی بین ۰ و  $\pi$  یباشد. عبارت داخل پرانتز سمت راست رابطه (۵) به جهت متناسب کردن مقدار  $\Delta x$  ابعاد مسأله است. شایان ذکر است برای پاسخگویی بهتر الگوریتم، در اوایل



شکل ۱. ایجاد ۴ نقطه جدید در فاز ۱

Fig. 1. Creating 4 new points in phase 1

$$x_{i_{new\ 4}}^{d=2m-1} = x_{i_{center}}^{d=2m-1} - a1 \times (a4 \times r) \quad (14)$$

$$x_{i_{new\ 4}}^{d=2m} = x_{i_{center}}^{d=2m} - a1 \times \sqrt{r^2 - (a4 \times r)^2}$$

مقدار  $a4, a3, a2, a1$  هر عدد تصادفی بین ۰ و ۱ میباشند. مقدار تابع ارزیاب در نقاط جستجوی ایجاد شده روی محیط دایره و داخل آن محاسبه می شود. سپس با استفاده از این اطلاعات چهار نقطه جستجوی دیگر ایجاد می شود. برای ایجاد نقاط جدید ابتدا نیمی از نقاط کنار هم روی محیط دایره که مجموع مقدار تابع ارزیاب آنها بهترین مقدار است تعیین می-شود، تأکید میشود که مقصود از بهترین مقدار تابع ارزیاب، در مسائل کمینه‌یابی کمترین مقدار آن و در مسائل بیشینه‌یابی بیشترین مقدار آن میباشند. پس از آن، جهت حرکت خروج از دایره، به سمت این نیمه در نظر گرفته می شود. از بین تمام نقاط جستجوی ایجاد شده داخل و روی محیط دایره چهار نقطه که بهترین وضعیت را دارا هستند انتخاب میشوند و با توجه به این که در جهت خروج از دایره هستند یا ورود، هر نقطه یک نقطه جستجوی جدید به دست خواهد داد. نحوه جهت‌یابی در شکل ۲ دیده میشود.

$$\Delta x = x_0 \cdot \left( \frac{i_{max\ 2}}{i-1} \right) \cdot \left( \frac{x_{max} - x_{min}}{2 \times i_{max\ 2}} \right) \quad (15)$$

$$\begin{cases} x_{i_{new}}^d = x_{i_{point}}^d + \Delta x \cdot dir & \text{If located in output direction} \\ x_{i_{new}}^d = x_{i_{center}}^d + \Delta x \cdot dir & \text{If located in input direction} \end{cases} \quad (16)$$

در این رابطه‌ها  $i_{max\ 2}$  حداکثر تعداد تکرار در فاز دوم و  $i$  نیز شمارنده گامهای فاز دوم میباشند. منظور از  $dir$  جهت است که در واقع یک عدد مثبت یا منفی میباشند که با توجه به توضیحی که پیشتر آمد، تشخیص داده میشود.

وضعیت نقاط جستجوی جدید به کمک تابع ارزیاب، ارزیابی می-شود و از بین این ۴ نقطه و ۴ نقطه برداشت شده از دایره، هر کدام که مقدار تابع ارزیاب بهتری داشته باشد، به عنوان بهترین نقطه این گام در نظر گرفته می شود و دایره برای گام بعد به این نقطه پرش میکند. این محاسبات تا زمانی که شرط توقف فاز دوم برقرار شود ادامه می یابد. در پایان بهترین نقطه آخرین گام، پاسخ بهینه مسأله خواهد بود. فلوچارت الگوریتم پرش دایره ها در شکل ۳ آورده شده است.

$$x_{i_{new}}^{d=2m-1} = x_{i_{center}}^{d=2m-1} + r_i \times \cos(\alpha) \quad (7)$$

$$x_{i_{new}}^{d=2m} = x_{i_{center}}^{d=2m} + r_i \times \sin(\alpha) \quad (8)$$

در این رابطه  $2m-1$  یانگر جهت (بعد) فرد و  $2m$  یانگر جهت (بعد) زوج میباشند.  $\alpha$  اویه نقطه موردنظر است و بایستی از مقادیر بازه  $[0, 2\pi]$  که به صورت مساوی به تعداد نقاط روی محیط تقسیم شده اند، گزینش شود. توصیه می شود در مسائلی که سرعت اهمیت بیشتری دارد، تعداد ۴ یا ۶ نقطه و برای مسائلی که دقت اهمیت بیشتری دارد ۲۴ تا ۳۰ نقطه روی محیط دایره در نظر گرفته شود.  $r_i$  ه در روابط بالا آمده است همان شعاع دایره در گام  $i$  از فاز دوم است که به شرح زیر به صورت خطی، در هر گام کاهش مییابد:

$$r_0 = 0.03(x_{max}^d - x_{min}^d) \quad (9)$$

$$r_i = r_0 - r_0 \times \frac{i-1}{i_{max\ 2}} \quad (10)$$

در این رابطه‌ها  $x_{max}^d$  و  $x_{min}^d$  به ترتیب حداکثر و حداقل مقادیر قابل برداشت برای متغیر  $d$ ، شعاع اولیه دایره در فاز دوم و  $r_i$  شعاع دایره در گام  $i$  از فاز دوم می باشد.  $i_{max\ 2}$  حداکثر تعداد گام در فاز دوم است. شایان ذکر است  $r_i$  در انتهای فاز اول به صفر می رسد، ولی فاز دوم را با حدوداً یک سوم مقدار اولیه فاز قبل آغاز می کند. این امر، همپوشانی دو فاز را سبب میشود که کمک شایانی به بهبود پاسخها می کند.

در الگوریتم پرش دایره ها تعداد نقاط جستجوی دایره مضربی از ۴ در نظر گرفته شده است. رابطه‌هایی که در ادامه می‌آید این نقاط را به گونه‌ای ایجاد میکند که در هر ربع دایره به تعداد مساوی نقطه جستجو داشته باشیم.

$$x_{i_{new\ 1}}^{d=2m-1} = x_{i_{center}}^{d=2m-1} + a4 \times (a1 \times r) \quad (11)$$

$$x_{i_{new\ 1}}^{d=2m} = x_{i_{center}}^{d=2m} + a4 \times \sqrt{r^2 - (a1 \times r)^2}$$

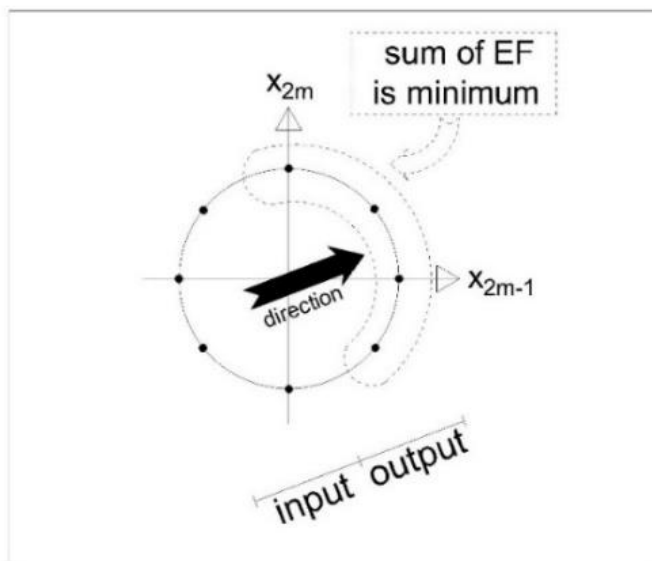
$$x_{i_{new\ 2}}^{d=2m-1} = x_{i_{center}}^{d=2m-1} + a3 \times (a2 \times r) \quad (12)$$

$$x_{i_{new\ 2}}^{d=2m} = x_{i_{center}}^{d=2m} - a3 \times \sqrt{r^2 - (a2 \times r)^2}$$

$$x_{i_{new\ 3}}^{d=2m-1} = x_{i_{center}}^{d=2m-1} - a2 \times (a3 \times r) \quad (13)$$

$$x_{i_{new\ 3}}^{d=2m} = x_{i_{center}}^{d=2m} + a2 \times \sqrt{r^2 - (a3 \times r)^2}$$





شکل ۲. تعیین جهت حرکت دایره در فاز ۲

Fig. 2. Evaluating movement direction of the circle in phase 2

### ۳- معیار همگرایی (شرط توقف)

علاوه بر تعداد گامها (تعداد تکرار محاسبات در هر فاز) که در ابتدای حل بایستی مشخص شود، چنانچه در ۱۰ گام متوالی تغییر چندانی در مقدار تابع ارزیاب مشاهده نگردد الگوریتم متوقف می شود. شکل ریاضی شرط توقف به صورت زیر می باشد:

$$\frac{EF_{i-10} - EF_i}{EF_{i-10}} \times 100 \leq 0.5, \quad i \geq 20 \quad (17)$$

### ۴- مسائل محک سازهای

مسئله های خرپای صفحه ای ۱۰ عضوی، خرپای فضایی ۲۵ عضوی و خرپای فضایی ۷۲ عضوی سه مسئله محک مشهور مهندسی میباشند. هدف این مسائل کمینه سازی وزن خرپا می باشد. بدیهی است وزن خرپا از رابطه زیر محاسبه می شود:

$$W = \sum_{i=1}^n \gamma_i A_i L_i \quad (18)$$

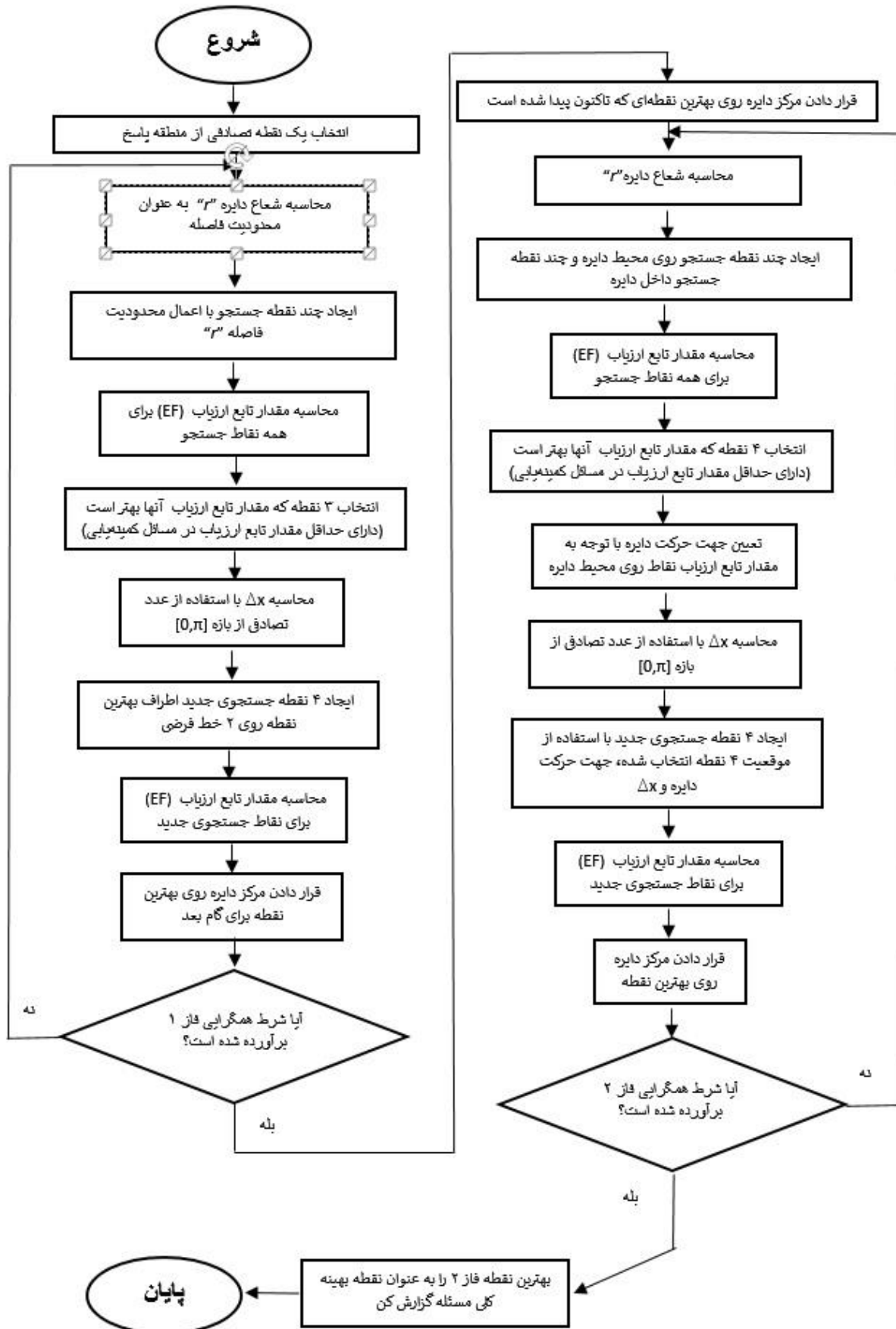
در این رابطه  $\gamma$  وزن مخصوص مصالح،  $A$  سطح مقطع میلهها و  $L$  طول میله های خرپا میباشد. سه مسئله یاد شده با الگوریتم پرش دایرهها حل شده و نتایجی که از ۲۰ بار اجرای مستقل برنامه به دست

آمده با نتایج ارائه شده در سایر مقالات مقایسه گردیده است. در ادامه نتایج به دست آمده در جدولی به تفکیک ارائه خواهد شد.

### ۴-۱- خرپای صفحه‌های ۱۰ عضوی

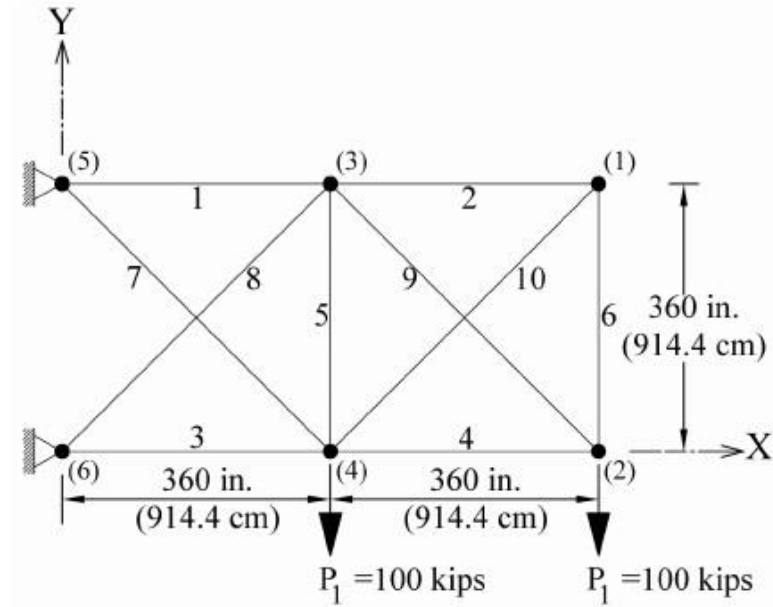
در مسأله خرپای صفحه‌های ۱۰ عضوی، مصالح دارای چگالی  $0.1 \text{ lb/in}^3$  و  $2.8 \text{ Kg/m}^3$  و مدول الاستیسیته  $0 \text{ msi}$  و  $0 \text{ GPa}$  (۹۵) می باشند. جابه‌جایی گره های خرپا و تنش اعضای خرپا به ترتیب به مقدارهای  $0.8 \text{ in} \pm$  و  $5 \text{ ksi} \pm$  و  $0.4 \text{ KN} \pm$  محدود میشود. دو بار  $PI$  با مقدار  $0.0$   $KN$  (۴،۸) به سازه اعمال میشود. حداقل و حداکثر سطح مقطع میله ها به ترتیب ۱  $in^2$  و ۵  $in^2$  می باشد. هندسه مربوط به این مسأله در شکل ۴ آورده شده است. در حل این مسأله به وسیله الگوریتم پرش دایره ها، تعداد نقاط جستجو در فاز اول ۱۷ عدد (۱ در مرکز دایره و ۱۶ پیرامون آن) و در فاز دوم ۳۲ عدد (۱۶ درون دایره و ۱۶ روی محیط دایره)، تعداد تکرار در هر فاز ۵۰ عدد می باشد.

مسأله خرپای ۱۰ عضوی پیشتر توسط اشमित و میورا [۲۱]، لی و جیم [۲۲]، کاوه و رحامی [۲۳]، لی و همکاران [۲۴] و ورعی و قاسمی [۲۵] حل شده است. نتایج به دست آمده از روش پرش دایره ها با نتایج ارائه شده توسط پژوهشگران یاد شده مقایسه می گردد و



شکل ۳. فلوجارت روش پرش دایره ها

Fig. 3. The flowchart of the Jump of Circles Optimization Method



شکل ۴. هندسه خربای ۱۰ عضوی

Fig. 4. The geometry of the 10 bars truss

جدول ۱. مقایسه نتایج خربای ۱۰ عضوی

Table 1. Comparing results for 10 bars truss

شماره عضو	اشمیت و میورا [21]	لی و جیم [22]	کاوه و رحامی [23]	لی و همکاران [24]	ورعی و قاسمی [25]	روش پیشنهادی
1	30.67	30.15	30.668	30.569	30.507	30.513
2	0.1	0.102	0.1	0.1	0.1	0.1
3	23.76	22.71	22.872	22.974	23.302	23.289
4	14.59	15.27	15.345	15.148	15.195	15.203
5	0.1	0.102	0.1	0.1	0.1	0.1
6	0.1	0.54	0.464	0.547	0.544	0.531
7	8.57	7.54	7.48	7.493	7.461	7.489
8	21.07	21.56	20.965	21.159	21.113	21.114
9	20.96	21.45	21.703	21.556	21.414	21.412
10	0.1	0.1	0.1	0.1	0.1	0.1
وزن (پوند)	5076.85	5057.88	5061.9	5061.03	5060.99	5060.64
تعداد آنالیز	بیان نشده	400,000	بیان نشده	150,000	13,800	5,662
تخطی قیود	0	0.001	0	0	0	0

عدد (۱ در مرکز دایره و ۱۲ پیرامون آن) و در فاز دوم ۲۴ عدد (۱۲ درون دایره و ۱۲ روی محیط دایره)، تعداد تکرار در هر فاز ۲۵ عدد می باشد.

مسئله خرپای ۲۵ عضوی پیشتر توسط لی و جیم [۲۲]، سانمز [۲۶]، کاوه و همکاران [۲۷] و ورعی و قاسمی [۲۵] حل شده است. نتایج به دست آمده از روش پرش دایرهها با نتایج ارائه شده توسط پژوهشگران یاد شده مقایسه می گردد و در جدول ۴ ارائه میشود.

از تعداد آنالیزهای ارائه شده در جدول ۴ تعداد ۹۶۱ مربوط به فاز اول و تعداد ۱۲۳۳ مربوط به فاز دوم می باشد. رسیدن به پاسخ مطلوب با تخطی قیود صفر و فراخوانی تابع بسیار کم، برتری نسبی روش پیشنهادی را به روشنی نمایان می سازد. تاریخچه همگرایی روش یادشده نیز در شکل ۷ آورده شده است.

#### ۴-۳- خرپای فضایی ۷۲ عضوی

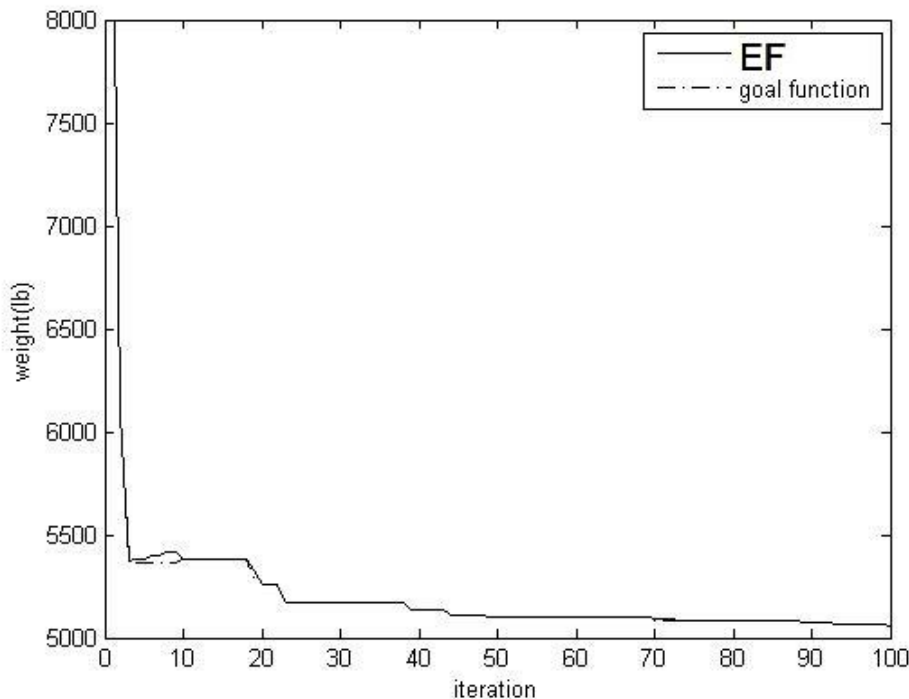
در مسئله خرپای فضایی ۷۲ عضوی، مصالح دارای چگالی  $1.0 \frac{lb}{in^3}$  و  $0.000786 \frac{Kg}{m^3}$  و مدول الاستیسیته  $100 \text{ GPa}$  و  $7.17 \times 10^{10} \text{ msi}$  (۹۵). میباشند. جابه‌جایی گره های خرپا و تنش اعضای خرپا به

در جدول ۱ ارائه می شود.

از تعداد آنالیزهای ارائه شده در جدول ۱ تعداد ۲۶۹۰ مربوط به فاز اول و تعداد ۲۹۷۲ مربوط به فاز دوم می باشد. رسیدن به پاسخ بهتر با تعداد فراخوانی تابع بسیار کمتر، برتری نسبی روش پیشنهادی را به روشنی نمایان می سازد. تاریخچه همگرایی روش یاد شده نیز در شکل ۵ آورده شده است.

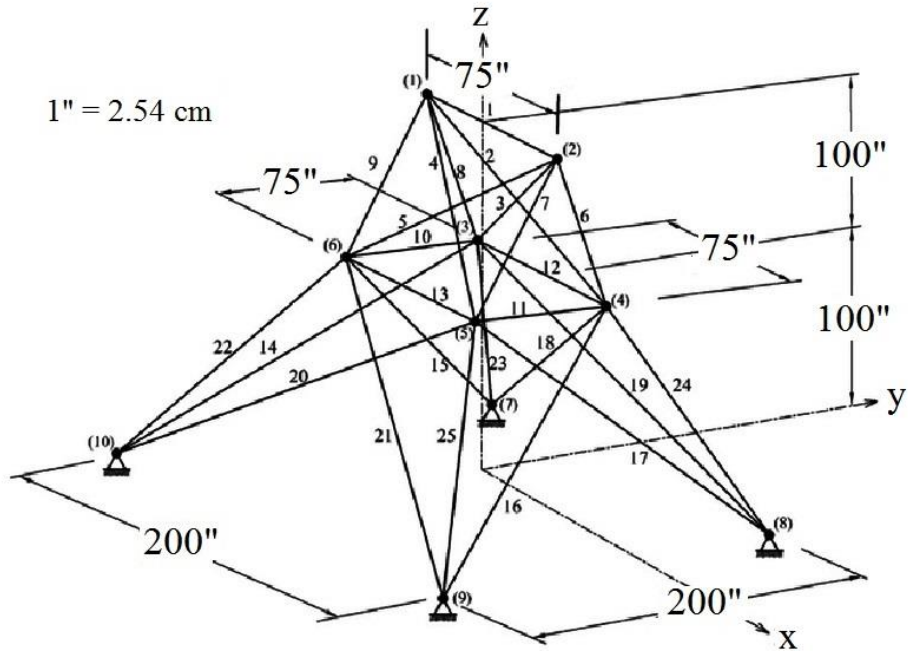
#### ۴-۲- خرپای فضایی ۲۵ عضوی

در مسئله خرپای فضایی ۲۵ عضوی، مصالح دارای چگالی  $1.0 \frac{lb}{in^3}$  و  $0.000786 \frac{Kg}{m^3}$  و مدول الاستیسیته  $100 \text{ GPa}$  و  $7.17 \times 10^{10} \text{ msi}$  (۹۵). میباشند. جابه‌جایی گرههای خرپا به مقدار  $0.35 \text{ in} \pm \text{cm}$  ( $\pm 0.0138 \text{ m}$ ) محدود می شود. حداقل و حداکثر تنش مجاز برای اعضای مختلف در جدول ۲ بیان شده است. دو ترکیب بار متفاوت که در جدول ۳ بیان شده است به سازه اعمال می شود. حداقل و حداکثر سطح مقطع میله ها به ترتیب  $0.1 \text{ in}^2$  و  $1.0 \text{ in}^2$  می باشد. هندسه مربوط به این مسئله در شکل ۶ آورده شده است. در حل این مسئله به وسیله الگوریتم پرش دایره ها، تعداد نقاط جستجو در فاز اول ۱۳



شکل ۵. تاریخچه همگرایی خرپای ۱۰ عضوی

Fig. 5. Convergence history for 10 bars truss



شکل ۶. هندسه خرپای ۲۵ عضوی

Fig. 6. The geometry of the 25 bars truss

جدول ۲. محدودیت تنش اعضا در خرپای ۲۵ عضوی (ksi)

Table 2. Elements' Tension/compression limit for 25 bars truss (ksi)

شماره عضو	محدودیت تنش فشاری	محدودیت تنش کششی
1	35.092	40
2-5	11.59	40
6-9	17.305	40
10-11	35.092	40
12-13	35.092	40
14-17	6.759	40
18-21	6.959	40
22-25	11.082	40

جدول ۳. شرایط بارگذاری خرپای ۲۵ عضوی (kips)

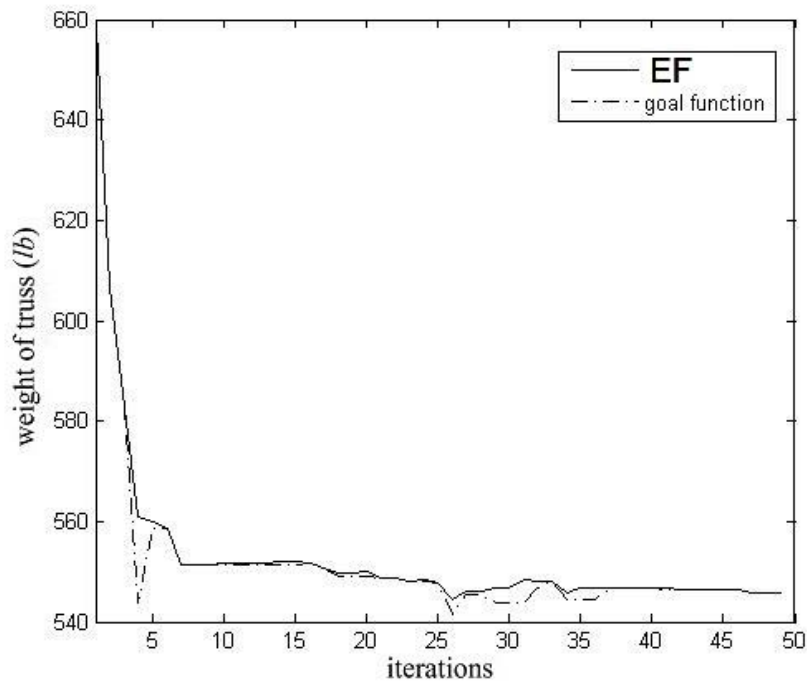
Table 3. Loading conditions for 25 bars truss (kips)

نوع بارگذاری	گره	جهت		
		x	y	z
1	1	1	10	-5
	2	0	10	-5
	3	0.5	0	0
	5	0	20	-5
	6	0	-20	-5
	2			

جدول ۴. مقایسه نتایج خرابی ۲۵ عضوی

Table 4. Comparing results for 25 bars truss

شماره عضو	لی و جیم [22]	سانمز [26]	کاوه و همکاران [27]	ورعی و قاسمی [25]	روش پیشنهادی
1	0.047	0.011	0.01	0.01	0.0122
2-5	2.022	1.979	1.91	2.006	1.8514
6-9	2.95	3.003	2.798	2.961	3.0353
10-11	0.01	0.01	0.01	0.01	0.0100
12-13	0.014	0.01	0.01	0.01	0.0100
14-17	0.688	0.69	0.708	0.687	0.6762
18-21	1.657	1.679	1.836	1.676	1.7607
22-25	2.663	2.652	2.645	2.668	2.6410
وزن (پوند)	544.38	545.19	545.09	545.19	545.12
تعداد آنالیز	15,000	300,000	17,500	5,240	2,194
تخطی قیود	0.0715	0.058	0.0202	0	0



شکل ۷. تاریخچه همگرایی خرابی ۲۵ عضوی

Fig. 7. Convergence history for 25 bars truss

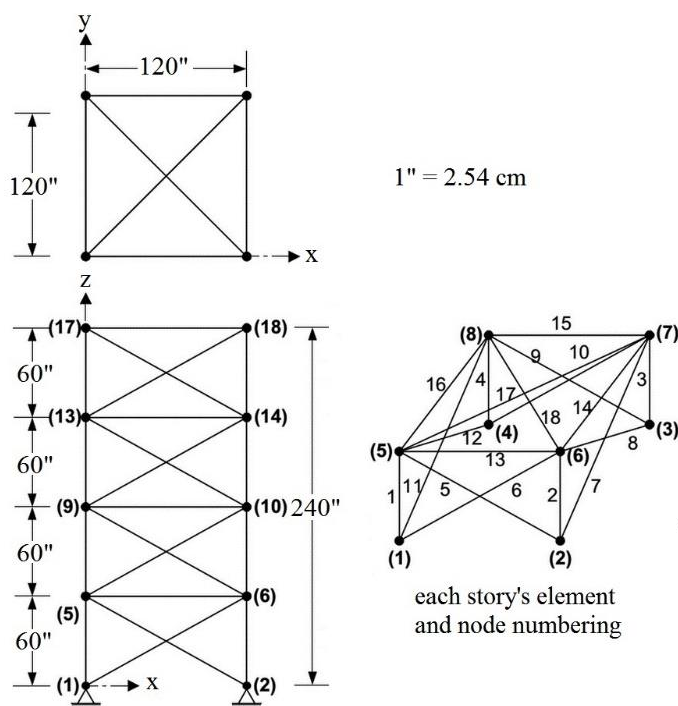
خان و همکاران [۲۹]، اربیتور و همکاران [۳۰]، لی و جیم [۲۲] و دیجرتکین [۳۱] حل شده است. نتایج به دست آمده از روش پرش دایره ها با نتایج ارائه شده توسط پژوهشگران یاد شده مقایسه می گردد و در جدول ۶ ارائه می شود.

از تعداد آنالیزهای ارائه شده در جدول ۶ تعداد ۱۷۲۹ مربوط به فاز اول و تعداد ۱۵۵۱ مربوط به فاز دوم می باشد. با توجه به اینکه روش ارائه شده با تعداد فراخوانی تابع بسیار کمتری نسبت به سایر روشها به پاسخ مطلوب رسیده است و همچنین تخطی قیود صفر می باشد، برتری نسبی روش پیشنهادی به روشنی قابل مشاهده است. تاریخچه

ترتیب به مقدارهای  $2.5 \text{ } Cm \pm in$  و  $5 \text{ } ksi \pm MPa$  ( $\pm 64$ ) محدود می شود. دو ترکیب بار متفاوت که در جدول ۵ بیان شده است به سازه اعمال میشود. حداقل و حداکثر سطح مقطع میله ها به ترتیب ۱.  $in^2$  و  $in^2$  می باشد. هندسه مربوط به این مسأله در شکل ۸ آورده شده است. در حل این مسأله به وسیله الگوریتم پرش دایرهها، تعداد نقاط جستجو در فاز اول ۱۷ عدد (۱ در مرکز دایره و ۱۶ پیرامون آن) و در فاز دوم ۲۴ عدد (۱۲ درون دایره و ۱۲ روی محیط دایره)، تعداد تکرار در هر فاز ۲۷ عدد می باشد. مسأله خرپای ۷۲ عضوی پیشتر توسط اشمیت و فرشی [۲۸]،

جدول ۵. شرایط بارگذاری خرپای ۷۲ عضوی (kips)  
Table 5. Loading conditions for 72 bars truss (kips)

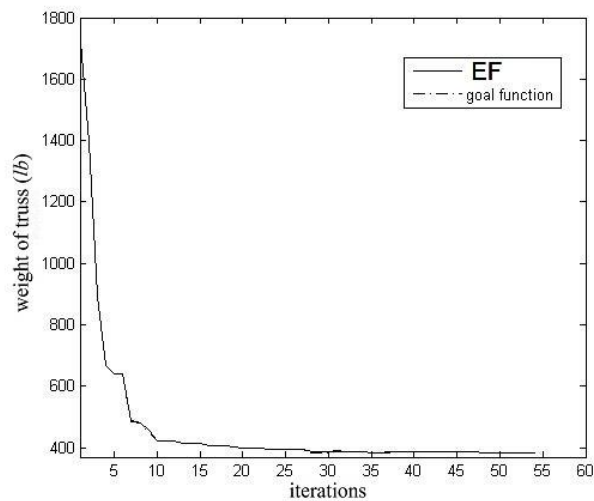
جهت			گره	نوع بارگذاری
z	y	x		
-5	5	5	17	1
-5	0	0	17	2
-5	0	0	18	
-5	0	0	19	
-5	0	0	20	



شکل ۸. هندسه خرپای ۷۲ عضوی  
Fig. 8. The geometry of the 72 bars truss

جدول ۶. مقایسه نتایج خرابای ۷۲ عضوی  
**Table 6. Comparing results for 72 bars truss**

شماره عضو	اشمیت و فرشی [28]	خان و همکاران [29]	اربتور و همکاران [30]	لی و جیم [22]	دیجرتکین [31]	روش پیشنهادی
1-4	2.078	1.859	1.755	1.790	1.860	1.8320
5-12	0.503	0.526	0.505	0.521	0.521	0.5297
13-16	0.100	0.100	0.105	0.100	0.100	0.1000
17-18	0.100	0.100	0.155	0.100	0.100	0.1000
19-22	1.107	1.253	1.155	1.229	1.271	1.3772
23-30	0.579	0.524	0.585	0.522	0.509	0.4972
31-34	0.100	0.100	0.100	0.100	0.100	0.1000
35-36	0.100	0.100	0.100	0.100	0.100	0.1000
37-40	0.264	0.581	0.460	0.517	0.485	0.5147
41-48	0.548	0.527	0.530	0.504	0.501	0.5551
49-52	0.100	0.100	0.120	0.100	0.100	0.1000
53-54	0.151	0.158	0.165	0.101	0.100	0.1000
55-58	0.158	0.152	0.155	0.156	0.168	0.1639
59-66	0.594	0.561	0.535	0.547	0.584	0.5271
67-70	0.341	0.438	0.480	0.442	0.433	0.4494
71-72	0.608	0.532	0.520	0.590	0.520	0.5327
وزن (پوند)	389.08	387.88	387.04	379.71	381.33	384.41
تعداد آنالیز	بیان نشده	بیان نشده	بیان نشده	20,000	13,742	3,280
تخطی قیود	0.8457	0.0326	0.1256	0.0756	0.0116	0



شکل ۹. تاریخچه همگرایی خرابای ۷۲ عضوی  
**Fig. 9. Convergence history for 72 bars truss**



به جواب مسأله را بسیار دشوار می کند. نمودار سطحی مربوط به حالت دو بعدی تابع کین در شکل ۱۰ آمده است. هندسه مربوط به این مسأله در شکل ۴ آورده شده است. در حل این مسأله به وسیله الگوریتم پرش دایرهها، تعداد نقاط جستجو در فاز اول ۱۷ عدد (۱ در مرکز دایره و ۱۶ پیرامون آن) و در فاز دوم ۳۲ عدد (۱۶ درون دایره و ۱۶ روی محیط دایره)، تعداد تکرار در هر فاز ۲۸۰ عدد می باشد. از آخرین محققینی که تابع یاد شده را با روش خود حل کرده اند می توان به انگ و کین [۳۲] و سالار و همکاران [۳۳] اشاره نمود. نتایج به دست آمده از الگوریتم پرش دایرهها برای ۲۰ مرتبه اجرای مستقل برنامه، برای تابع کین ۲۰ بعدی با نتایج ارائه شده توسط دانشمندان یاد شده مقایسه گردیده و در جدول ۷ ارائه شده است. تاریخچه همگرایی پاسخ ۰/۸۰۰۲ در شکل ۱۱ ارائه شده است. به منظور افزایش سرعت حل مسأله بایستی تعداد فراخوانی تابع را کاهش داد. بدیهی است که افزایش سرعت در صورتی که دقت موردنیاز مسأله را تأمین کند، صرفه جویی اقتصادی را نتیجه می دهد. روش پرش

همگرایی روش یاد شده نیز در شکل ۹ آورده شده است.

### ۵- مسأله محک ریاضی (بیشینه یابی)

تابع پر دست انداز کین یک مسأله محک مشهور است که توانایی روشهای بهینه سازی در حل مسأله را به خوبی به چالش میکشد. معادله تابع کین به صورت زیر است:

$$abs \left[ \sum_{i=1}^d \cos^4(x_i) - 2 \prod_{i=1}^d \cos^2(x_i) \right] / \sqrt{\sum_{i=1}^d i x_i^2} \quad (19)$$

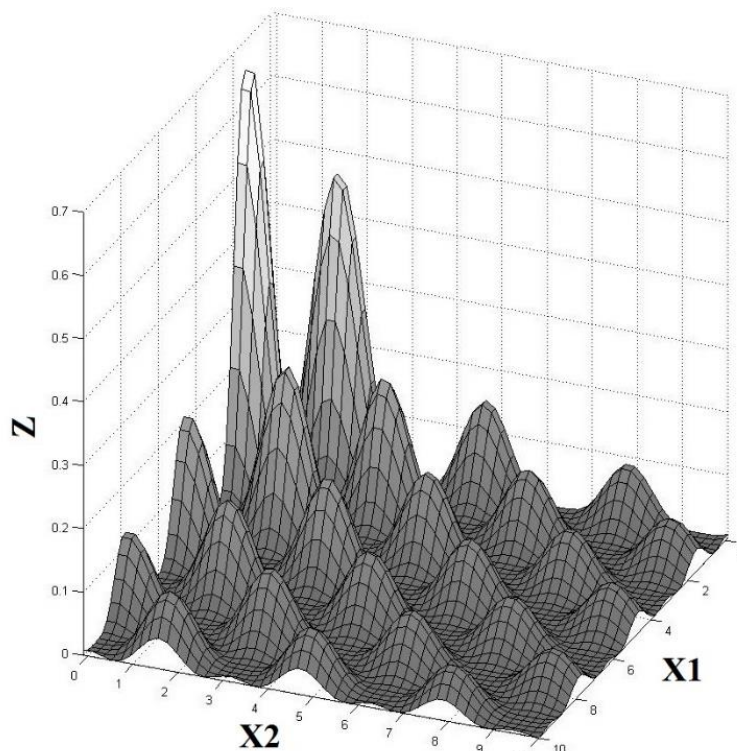
Subject to:

(20)

$$\prod_{i=1}^d x_i > 0.75, \quad \sum_{i=1}^d x_i < \frac{15d}{2}$$

$$0 \leq x_i \leq 10, \quad i = 1, 2, \dots, d$$

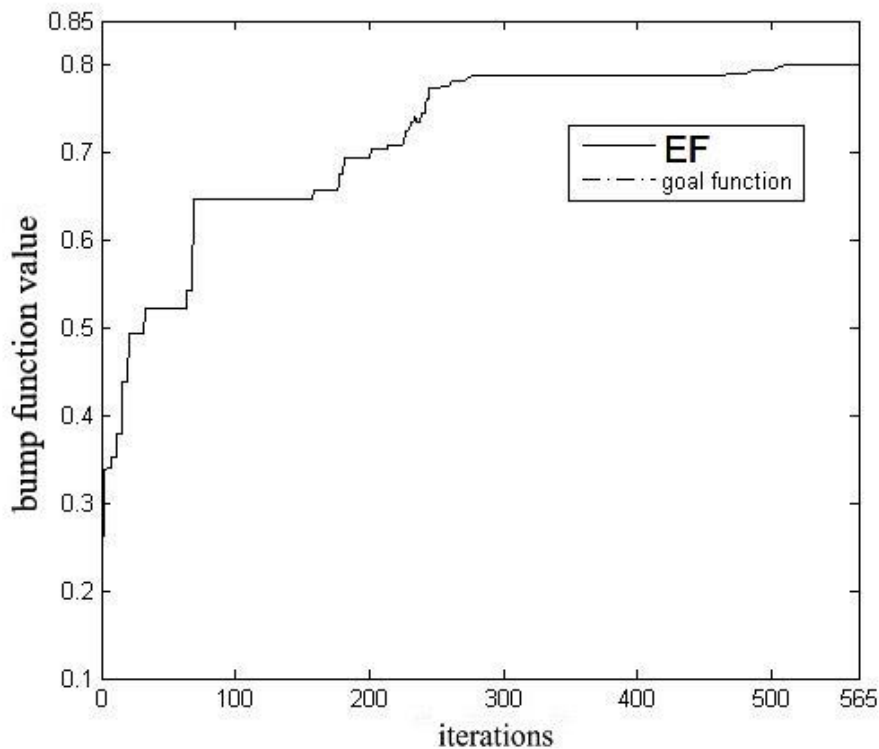
در این روابط  $d$  عدد تابع کین است. هرچه بعد تابع افزایش یابد، دست اندازهای سطح تابع افزایش زیادی خواهد یافت که رسیدن



شکل ۱۰. نمودار سطحی تابع کین ۲ بعدی  
Fig. 10. Surface plot of 2D Keane function

جدول ۷. مقایسه نتایج تابع کین ۲۰ بعدی  
**Table 7. Comparing results for 20D Keane function**

انحراف معیار	روش پیشنهادی			[33]	[32]	مقدار
	بدترین	میانگین	بهترین			
0.0231	0.7021	0.7464	0.8002	0.7917	0.738	تعداد انالیز
	58,600			92500	100,000	



شکل ۱۱. تاریخچه همگرایی تابع کین ۲۰ بعدی برای پاسخ ۰/۸۰۰۲  
**Fig. 11. 20D Keane function's Convergence history for response 0.8002**

مسئله سیلندر فشاری دارای دو متغیر پیوسته و دو متغیر گسسته می باشد. هندسه سیلندر فشاری در شکل ۱۳ آمده است. متغیرهای طراحی شامل ضخامت پوسته  $(T_s=x_1)$ ، ضخامت سر کروی  $(T_h=x_2)$ ، شعاع پوسته سیلندر  $(R=x_3)$  و طول پوسته  $(L=x_4)$  می باشند. متغیرهای  $x_1$  و  $x_2$  بایستی ضربی از ۰/۰۶۲۵ باشند. تابع هدف و قیدها به شرح زیر هستند:

$$\min f(\mathbf{X}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \quad (21)$$

دایره ها که در این مقاله ارائه شده است می تواند با تعداد فراخوانی تابع بسیار کمتری به پاسخی قابل قبول برسد. نتایج مربوط به حالت با فراخوانی تابع کم برای تابع کین ۲۰ بعدی در جدول ۸ و تاریخچه همگرایی بهترین جواب آن در شکل ۱۲ ارائه شده است.

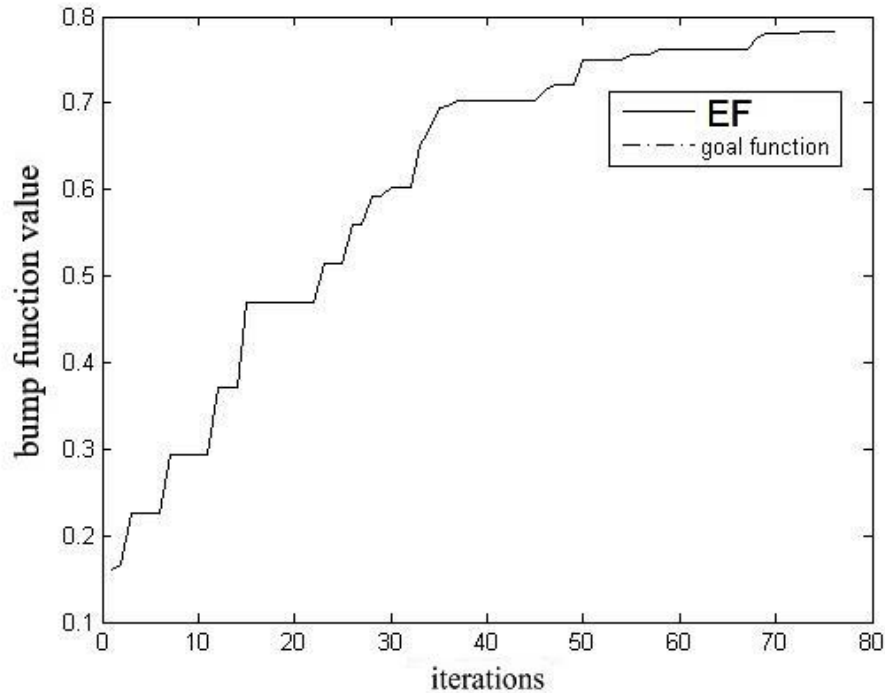
#### ۶- مسئله محک مکانیکی

هدف این مسئله کاهش هزینه ساخت یک سیلندر فشاری است.

جدول ۸. نتایج تابع کین ۲۰ بعدی با آنالیز کم

Table 8. Results for 20D Keane function- low number of analysis

انحراف معیار	بدترین	میانگین	بهترین	مقدار
0.6325	0.7062	0.7062	0.783	تعداد آنالیز
26,084	11,264	11,264	6,908	



شکل ۱۲. تاریخچه همگرایی تابع کین ۲۰ بعدی - تعداد آنالیز کم

Fig. 12. 20D Keane function's Convergence history-low number of analysis

مسأله طراحی بهینه سیلندر فشاری پیشتر توسط کونلو و مونتس به کمک GA [۳۴]، هی و ونگ به کمک CPSO [۳۵]، گائو و هایلو به کمک PSO [۳۶] و محلاتی و همکاران به کمک IMOEA [۳۷] حل شده است. نتایج به دست آمده از روش پرش دایره‌ها با نتایج ارائه شده توسط پژوهشگران یاد شده مقایسه می‌گردد و در جدول ۹ ارائه می‌شود. در حل این مسأله به وسیله الگوریتم پرش دایره‌ها، تعداد نقاط جستجو در فاز اول ۶۱ عدد (۱ در مرکز دایره و ۶۰ پیرامون آن) و در فاز دوم ۴۰ عدد (۲۰ درون دایره و ۲۰ روی محیط دایره)، تعداد تکرار در هر فاز ۵۰ عدد می‌باشد. از تعداد آنالیزهای ارائه شده در جدول ۹ تعداد ۳۸۱۰ مربوط به فاز اول و تعداد ۲۸۱۰ مربوط به فاز دوم می‌باشد. با توجه به اینکه

$$g_1(\mathbf{X}) = -x_1 + 0.0193x_3 \leq 0 \quad (22)$$

$$g_2(\mathbf{X}) = -x_2 + 0.00954x_3 \leq 0 \quad (23)$$

$$g_3(\mathbf{X}) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0 \quad (24)$$

$$g_4(\mathbf{X}) = x_4 - 240 \leq 0 \quad (25)$$

$$1 \times 0.0625 \leq x_1, x_2 \leq 99 \times 0.0625 \quad (26)$$

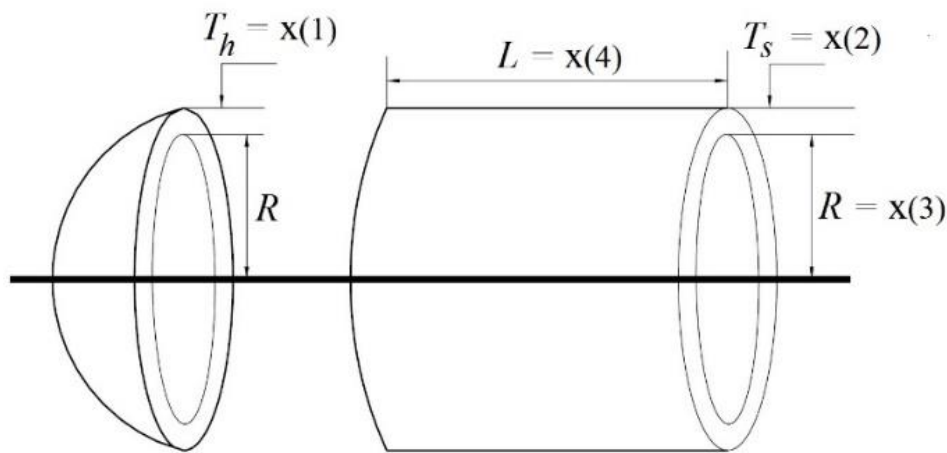
$$10 \leq x_3, x_4 \leq 200 \quad (27)$$

و آخر هر فاز در جدول ۱۱ آورده شده است. در این مسأله زمان نیز بررسی شده است. مشخصات رایانه مورد استفاده سی پی یو Ci7، رم 8GB و گرافیک 8GB می باشد. با این مشخصات تعداد 6620 آنالیز گزارش شده در مدت زمان 0/12 ثانیه انجام شده است و این بدان معناست که برای هر آنالیز حدود 5-10 ثانیه زمان لازم است.

روش ارائه شده با تعداد فراخوانی تابع بسیار کمتری نسبت به سایر روشها به پاسخ مطلوب رسیده است و همچنین تخطی قیود صفر می باشد، برتری نسبی روش پیشنهادی به روشنی قابل مشاهده است. نتایج آماری مربوط به روش پیشنهادی نیز با روش های بیان شده مقایسه گردیده و در جدول ۱۰ آورده شده است که برتری روش پیشنهادی قابل مشاهده است.

### ۷- نتیجه گیری

به منظور روشن تر شدن عملکرد الگوریتم، نتایج گام اول و دوم



شکل ۱۳. هندسه سیلندر فشاری

Fig. 13. The geometry of the pressure vessel

جدول ۹. نتایج مقایسه‌ای مخزن تحت فشار

Table 9. Comparing results for pressure vessel

روش پیشنهادی	IMOEA	PSO	CPSO	GA	
0.8125	0.8125	0.8125	0.8125	0.8125	$x_1$
0.4375	0.4375	0.4375	0.4375	0.4375	$x_2$
42.0984	42.0509	42.0984	42.0913	42.0974	$x_3$
176.6372	177.2305	176.6366	176.7465	176.654	$x_4$
-8.80E-07	-9.16E-04	-8.80E-07	-0.0001	-2.02E-05	$g_1$
-0.0376	-0.035	-0.0376	-0.0376	-0.0376	$g_2$
-0.218	-24.819	3.1227	-118.7688	-24.7594	$g_3$
-63.3628	-62.769	-63.3634	-63.2535	-63.346	$g_4$
6059.7208	6059.8362	6059.7068	6061.0836	6059.9456	$f_{min}$
6620	12000	60000	240000	80000	تعداد آنالیز

جدول ۱۰. نتایج آماری روش های مختلف برای مسأله مخزن تحت فشار  
**Table 10. Statistical results of the different method for pressure vessel**

تعداد آنالیز	انحراف معیار	بدترین	میانگین	بهترین	
80000	130.9267	6469.3220	6177.25	6059.9460	کوتلو و مونتس [34]
240000	86.4545	6363.8040	6147.13	6061.0780	هی و ونگ [35]
60000	12.2718	بیان نشده	6066.03	6059.7068	گائو و هایلو [36]
12000	4.3485	6069.9780	6060.77	6059.8362	محللاتی و همکاران [37]
6620	2.9563	6064.4256	6060.323	6059.7208	روش پیشنهادی

جدول ۱۱. نتایج نمونه چند گام از مسأله مخزن تحت فشار  
**Table 11. Few steps' results for pressure vessel**

تعداد آنالیز	تابع ارزیابی	تابع هدف	$x_4$	$x_3$	$x_2$	$x_1$	
77	12674236.6	12674236.6	127.80	49.72	75.11	92.79	فاز ۱ - گام ۱
75	636177.79	636177.79	112.31	48.82	57.10	77.87	فاز ۱ - گام ۲
76	14065.19	14065.19	50.97	46.23	10.01	18.08	فاز ۱ - گام نهایی
57	13891.14	13891.14	120.34	42.12	8.06	16.12	فاز ۲ - گام ۱
57	9814.18	9814.18	180.66	40.87	7.02	14.23	فاز ۲ - گام ۲
55	6059.7208	6059.7208	176.6372	42.0984	0.4375	0.8125	فاز ۲ - گام نهایی

در پایان، کارایی روش پیشنهادی به کمک سه مسأله مقید مهندسی و مسأله ریاضی بیشینه یابی تست گردید. بهبود پاسخهای روش پیشنهادی نسبت به سایر روش های موجود در ادبیات به وضوح قابل مشاهده است. از این رو، استفاده از روش بهینه سازی پرش دایره ها برای حل مسائل بهینه سازی مقید و نامقید به کاربران توصیه می شود.

### مراجع

- [1] Holland JH. Genetic algorithms. Scientific American, 267 (1992) 66-72.
- [2] Rao RV, Savsani VJ, Vakharia DP. Teaching-learning-based optimization: an optimization method for continuous non-linear large scale problems. Information Science, 183(1) (2012) 1-15.
- [3] Geem ZW, Kim JH, Loganathan G. A new heuristic

در این مقاله روش بهینه سازی پرش دایره ها معرفی گردید. این روش، در حقیقت روشی فرااکتشافی است که با محدودیت های هندسی ترکیب شده است. الگوریتم پیشنهادی شامل دو فاز می باشد. فاز اول اکتشاف بهینه ترین محدوده و فاز دوم بهره برداری از اکتشاف است. واضح است که هر فاز یکی از اصول مهم بهینه سازی را برعهده دارد. افزون بر آن برای پاسخ گیری بهتر، بین دو فاز همپوشانی برقرار شده است. همچنین برداشت نقاط از قسمت های مختلف فضای جستجو به کمک کاربرد فرااکتشافی به خوبی انجام می پذیرد. در روش بهینه سازی پرش دایره ها سعی بر آن بوده که اصول و ارکان مهم بهینه سازی در زمان مناسب و به نحوی مطلوب با هم ترکیب شوند که در حل مسائل تعداد آنالیزها کاهش یابد. وضعیت نقاط جستجو در هر فاز به کمک مفهومی تحت عنوان تابع ارزیابی، بررسی گردید.

- [17] Erol OK, Eksin I. A new optimization method: big bang–big crunch. *Advances in Engineering Software*, 37(2) (2006) 106-111.
- [18] Formato RA. Central force optimization: A new metaheuristic with applications in applied electromagnetics. *Progress in Electromagnetics Research*, 77 (2007) 425-491.
- [19] Rashedi E, Nezamabadi Pour H, Saryazdi S. GSA: a gravitational search algorithm. *Information Science*, 179(13) (2009) 2232-2248.
- [20] Naderi, A., Sohrabi, M.R., Ghasemi, M.R., Dizangian B., Total and Partial Updating Technique: A Swift Approach for Cross-Section and Geometry Optimization of Truss Structures. *KSCE J Civ. Eng.*, 24 (2020) 1219–1227.
- [21] Schmit Jr. L.A., Miura H., Approximation concepts for efficient structural synthesis. US National Aeronautics and Space Administration, 1976.
- [22] Lee K.S., Geem Z.W., A new structural optimization method based on the harmony search algorithm. *Computers & Structures*, 82(9-10) (2004) 781-798.
- [23] Kaveh A., Rahami H., Analysis, design and optimization of structures using force method and genetic algorithm. *International Journal for Numerical Methods in Engineering*, 65(10) (2006) 1570-1584.
- [24] Li L.J., Huang Z.B., Liu F., A heuristic particle swarm optimization method for truss structures with discrete variables. *Computers & Structures*, 87(7-8) (2009) 435-443.
- [25] Varae H., Ghasemi M.R., Engineering optimization based on ideal gas molecular movement Algorithm. *Engineering with Computers*, 33(1) (2016) 71-93.
- [26] Sonmez M. Artificial Bee Colony algorithm for optimization of truss structures. *Applied Soft Computing*, 11(2) (2011) 2406-2418.
- [27] Kaveh A., Sheikholeslami R., Talatahari S., Keshvari-Ilkhichi M., Chaotic swarming of particles: A new method for size optimization of truss structures. *Advances in Engineering Software*, 67(2014) 136-147.
- [28] Schmit Jr. L.A., Farshi B., Some Approximation Concepts for Structural Synthesis. *AIAA Journal*, 12 (5) optimization algorithm: harmony search. *Simulation*, 76(2) (2001) 60-68.
- [4] Fogel D. *Artificial intelligence through simulated evolution*. Wiley-IEEE Press, 2009.
- [5] He S, Wu Q, Saunders J. A novel group search optimizer inspired by animal behavioral ecology. *Proceedings of the 2006 IEEE congress on evolutionary computation*, (2006) 1272–1278.
- [6] Rechenberg I. *Evolutionsstrategien*. Springer Berlin Heidelberg, (1978) 83-114.
- [7] J.R. Koza. “Genetic programming”. 1992.
- [8] Simon D. Biogeography-based optimization. *IEEE Transactions on Evolutionary Computation*, 12(6) (2008) 702-713.
- [9] Dasgupta D, Zbigniew M, editors. *Evolutionary algorithms in engineering applications*. Springer Science & Business Media, 2013.
- [10] Kennedy J, Eberhart R. Particle swarm optimization. *Proceedings of the 1995 IEEE international conference on neural networks*, (1995) 1942–1948.
- [11] Dorigo M, Birattari M, Stutzle T. Ant colony optimization. *IEEE Computational Intelligence*, 1(4) (2006) 28-39.
- [12] Askarzadeh A, Rezazadeh A. A new heuristic optimization algorithm for modeling of proton exchange membrane fuel cell: bird mating optimizer. *International Journal of Energy Research*, 37(10) (2012) 1196-1204.
- [13] Pan W-T. A new fruit fly optimization algorithm: taking the financial distress model as an example. *Knowledge-Based Systems*, 26 (2012) 69-74.
- [14] Gandomi AH, Alavi AH. Krill Herd: a new bio-inspired optimization algorithm. *Communications in Nonlinear Science and Numerical Simulation*, 17(12) (2012) 4831-4845.
- [15] Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. *Science*, 220(4598) (1983) 671-680.
- [16] Webster B, Bernhard PJ. A local search optimization algorithm based on natural principles of gravitation. *Proceedings of the 2003 international conference on information and knowledge engineering (IKE'03)*, (2003) 255–261.

- [34] Coello C.A.C., Montes E.M., "Use of dominance-based tournament selection to handle constraints in genetic algorithms", *Intelligent Engineering Systems through Artificial Neural Network*, 11 (2001) 177-182.
- [35] He Q., Wang L., "An effective co-evolutionary particle swarm optimization for constrained engineering design problems", *Engineering Applications of Artificial Intelligence*, 20(1) (2007) 89-99.
- [36] Gao L., Hailu A., "Comprehensive learning particle swarm optimizer for constrained mixed-variable optimization problems", *International Journal of Computational Intelligence Systems*, 3(6) (2010) 832-842.
- [37] GHOHANI ARAB H., MAHALLATI RAYENI A., GHASEMI M., An Effective Improved Multi-objective Evolutionary Algorithm (IMOE) for Solving Constraint Civil Engineering Optimization Problems, *Teknik Dergi*, 32(2) (2021).
- (1974) 692-699.
- [29] Khan M.R., Willmert K.D., Thornton W.A., An Optimality Criterion Method for Large-Scale Structures. *AIAA Journal*, 17(7) (1979) 753-761.
- [30] Erbatur F., Hasancebi O., Tutuncu I., Kilic H., Optimal design of planar and space structures with genetic algorithms. *Computers & Structures*, 75(2) (2000) 209-224.
- [31] Degertekin S.O., Improved harmony search algorithms for sizing optimization of truss structures. *Computers & Structures*, 92-93 (2012) 229-241.
- [32] Ong Y.S., Keane A.J., Meta-Lamarckian Learning in Memetic Algorithms. *IEEE Transactions on Evolutionary Computation*, 8(2) (2004) 99-110.
- [33] Salar M, Ghasemi M R, Dizangian B. A FAST GA-BASED METHOD FOR SOLVING TRUSS OPTIMIZATION PROBLEMS. *International Journal of Optimization in Civil Engineering*, 6(1) (2016) 101-114.

چگونه به این مقاله ارجاع دهیم

N. Haji Aghajanzpour, M. R. Ghasemi, H. Ghohani Arab., (2021). *Jump of Circles: A New Way to Solve the Engineering Optimization Problems*. *Amirkabir J. Civil Eng.*, 53(5): 1917-1936.

DOI: [10.22060/ceej.2020.17227.6496](https://doi.org/10.22060/ceej.2020.17227.6496)



