सीमित समय में वाहन रूटिंग समस्या के लिए जेनेटिक रूप से ट्यून किए गए जुगुनू एल्गोरिथ्म

Genetically Tuned Firefly Algorithm for Vehicle Routing Problem with Time Windows

अपूर्व मिश्रा[1*], आदित्य दीक्षित[2], अनुपम शुक्ला[1]
Apoorva Mishra[1*], Aditya Dixit[2], Anupam Shukla[1]
[1]भारतीय सूचना प्रौद्योगिकी संस्थान, पुणे, भारत
[1]Indian Institute of Information Technology, Pune, India
[2]अटल बिहारी वाजपेयी–भारतीय सूचना प्रौद्योगिकी एवं प्रबंधन संस्थान, ग्वालियर, भारत
[2]Atal Bihari Vajpayee-Indian Institute of Information Technology and Management, Gwalior, India
apoorvamish1989@gmail.com[1*], a.dixit93@gmail.com[1], dranupamshukla@gmail.com[1]

सारांश :

प्रकृति–प्रेरित एल्गोरिथ्म (NIA) सबसे शक्तिशाली और मजबूत इष्टतमीकरण एल्गोरिथ्म में से एक है। फायर फ्लाई या जुगुनू एल्गोरिथ्म (FA) एक अपेक्षाकृत नया और लोकप्रिय एनआईए है जो जुगुनुओं के चमकते व्यवहार से प्रेरित है। जेनेटिक एल्गोरिथ्म (GA) विभिन्न प्रकार के इष्टतमीकरण समस्याओं को हल करने की क्षमता रखने वाले लोकप्रिय एल्गोरिथ्म में से एक है। इस शोध पत्र में, हम एक जेनेटिक रूप से ट्यून किए गए जुगुनू एल्गोरिथ्म (GFA) का प्रस्ताव करते हैं, जो दोनों एल्गोरिथ्म की ताकत का उपयोग करता है। प्रस्तावित दृष्टिकोण में, GA जुगुनू एल्गोरिथ्म के एक पैरामीटर की सेटिंग को स्वचालित करता है, और एफए इष्टतम समाधान खोजने की कोशिश करता है। जेनेटिक रूप से ट्यून किए गए जुगुनू एल्गोरिथ्म के प्रदर्शन के परीक्षण के लिए हम इसे सीमित समय में वाहन रूटिंग समस्या (VRPTW) के बेंचमार्क सोलोमन के डेटासेट के विभिन्न उदाहरणों पर लागू करते हैं। परिणाम दर्शाते हैं कि अधिकांश मामलों में जेनेटिक रूप से ट्यून किए गए जुगुनू एल्गोरिथ्म निकट–इष्टतम समाधान प्राप्त करता है।

Abstract

Nature-Inspired algorithms (NIAs) are one of the most powerful and robust optimization algorithms. Firefly Algorithm (FA) is a relatively new and popular NIA that is inspired by the flashing behavior of fireflies. Genetic Algorithm (GA) is one of the popular evolutionary algorithms having the potential to solve different kinds of optimization problems. In this paper, we propose a Genetically Tuned Firefly Algorithm (GFA), which utilizes the strengths of both these algorithms. In the proposed approach, GA automates the setting of a parameter of FA, and the FA tries to find the optimal solution. For testing the performance of the GFA, we apply it to various instances of the benchmark Solomon's dataset for Vehicle Routing Problem with Time Windows (VRPTW). The results indicate that the GFA achieves near-optimal solutions in a majority of the cases.

मुख्य शब्दः वाहन रूटिंग समस्या, जुगुनू एल्गोरिथ्म, जेनेटिक एल्गोरिथ्म, मेटा–ह्युरिस्टिक एल्गोरिथ्म।

Key Words: Vehicle Routing Problem, Firefly Algorithm, Genetic Algorithm, Metaheuristic Algorithms.

1. परिचय

इष्टतमीकरण (Optimization) समस्याओं को हल करने के लिए प्रकृति प्रेरित एल्गोरिथ्म का व्यापक रूप से उपयोग किया जाता है [1-3]। ऐसे एल्गोरिथ्म जो कि प्रकृति में पाए जाने वाले पशु पक्षियों या प्रकृति में होने वाली क्रियाओं से प्रेरित होते हैं उन्हें प्रकृति प्रेरित एल्गोरिथ्म कहा जाता है। जुगुनू एल्गोरिथ्म Xin-She Yang [1] द्वारा प्रस्तावित लोकप्रिय प्रकृति प्रेरित एल्गोरिथ्म है। यह एल्गोरिथ्म जुगुनुओं के जैविक व्यवहार से प्रेरित है। ऐसा पाया गया है कि एक जुगुनू अपने से ज्यादा उज्जवल जुगुनू कि ओर आकर्षित होता है और उसकी ओर बढ़ता है। परन्तु एक जुगुनू का दुसरे के प्रति आकर्षण दूरी बढ़ने के साथ कम होता जाता है। जुगुनुओं के इसी व्यहवार से प्रेरित है यह एल्गोरिथ्म।

यह एक प्रकार का स्टोकेस्टिक (Stochastic) एल्गोरिथ्म है जो जुगुनुओं के व्यवहार से अपनी प्रेरणा प्राप्त करता है। जुगुनू एल्गोरिथ्म के कई रूपों को भी इसके प्रदर्शन में सुधार करने के लिए प्रस्तावित किया गया है [4-5]। जुगुनू एल्गोरिथ्म के विकास से पहले, ग्लोवॉर्म (पंख रहित जुगुनुओं की प्रजाति) स्वार्म इष्टतमीकरण एल्गोरिथ्म [6] द्वारा पेश किया गया था, जो कि ग्लोवॉर्म द्वारा रोशनी के चमकने की घटना पर आधारित था। जेनेटिक एल्गोरिथ्म प्राकृतिक विकास की प्रक्रिया से प्रेरित है और आमतौर पर कई इष्टतमीकरण समस्याओं को हल करने के लिए उपयोग किया जाता है [7-10]। जेनेटिक एल्गोरिथ्म में शामिल प्रमुख ऑपरेटरः चयन, क्रॉसओवर और म्यूटेशन हैं। चयन प्रक्रिया में पापुलेशन में मौजूद क्रोमोसोम्स में से कुछ को उनकी फिटनेस के आधार पर क्रॉसओवर के लेया चुना जाता है। क्रॉसओवर में सामान्यतः एक बार में दो क्रोमोसोम्स भाग लेते है और नए क्रोमोसोम्स को जन्म देते हैं। म्युटेशन में क्रॉसओवर से जन्मे नए क्रोमोसोम्स में कुछ बेतरतीब तरीके से बदलाव किये जाते हैं। जेनेटिक एल्गोरिथ्म का गणितीय विश्लेषण [11-13] में प्रस्तुत किया गया है।

वाहन रूटिंग समस्या (VRP) एक जटिल इष्टतमीकरण समस्या है जो कई ग्राहकों की सेवा के लिए कई वाहनों के मार्गों के डिजाइन से संबंधित है [14]। VRPTW रसद प्रणालियों [15] में एक महत्वपूर्ण मुद्दा रहा है। VRPTW को हल करने के लिए विभिन्न छप्। को लागू किया गया है [15-16]। VRPTW को हल करने के लिए मेटा–ह्युरिस्टिक (Meta-heuristic) एल्गोरिथ्म के उपयोग पर एक हालिया सर्वेक्षण पत्र [19] में दिया गया है। मेटा–ह्युरिस्टिक एल्गोरिथ्म ऐसी जटिल समस्याएँ जिनका इष्टम समाधान निश्चित समय में नहीं निकला जा सकता उनका निकट इष्टतम समाधान निश्चित समय में प्रदान करते हैं।

हम जुगुनू एल्गोरिथ्म के नए प्रकार, अर्थात, जेनेटिक रूप से ट्यून किए गए जुगुनू एल्गोरिथ्म का प्रस्ताव करते हैं, जो जुगुनू एल्गोरिथ्म और जेनेटिक एल्गोरिथ्म दोनों की शक्ति का उपयोग करता है। हम जुगुनू एल्गोरिथ्म के एक पैरामीटर को ट्यून करने (अनुकूल बनाने) के लिए जेनेटिक एल्गोरिथ्म का उपयोग करते हैं। जेनेटिक रूप से ट्यून किए गए जुगुनू एल्गोरिथ्म, VRPTW के मानक डेटासेट के विभिन्न उदाहरणों पर लागू किया गया है ताकि इसके प्रदर्शन की तुलना सबसे अच्छे परिणामों से की जा सके।

शोध पत्र के शेष भाग को निम्नानुसार व्यवस्थित किया गया हैः खंड–2 जुगुनू एल्गोरिथ्म, जेनेटिक एल्गोरिथ्म और VRPTW के कामकाज से संबंधित बुनियादी अवधारणाओं का वर्णन करता है। खंड–3 में VRPTW पर जुगुनू एल्गोरिथ्म के प्रयोग की प्रक्रिया विस्तार से बताई गई है। खंड–4 में हम एक नए 'जेनेटिक रूप से ट्यून किए गए जुगुनू एल्गोरिथ्म' का प्रस्ताव करते हैं, जिसमें हमने जेनेटिक एल्गोरिथ्म का उपयोग करके जुगुनू एल्गोरिथ्म में शामिल एक पैरामीटर को ट्यून किया है। जेनेटिक रूप से ट्यून किए गए जुगुनू एल्गोरिथ्म के कामकाज के बारे में भी विस्तार से बताया गया है। खंड–5 VRPTW

के लिए बेंचमार्क डेटासेट (सोलोमन के डेटासेट) का वर्णन करता है, जिसका उपयोग किया जा रहा है। इस डेटासेट के विभिन्न उदाहरणों के लिए जेनेटिक रूप से ट्यून किए गए जुगुनू एल्गोरिथ्म लागू करके प्राप्त परिणामों को भी प्रस्तुत किया गया है। खंड–6 इस शोध का निष्कर्ष निकालता है और भविष्य में वृद्धि की गुंजाइश पर प्रकाश डालता है।

2. प्रारंभिक

2.1. जुगुनू एल्गोरिथ्म

जुगुनू एल्गोरिथ्म एक प्रकार का जैव–प्रेरित एल्गोरिथ्म है जो जुगुनुओं के व्यवहार से अपनी प्रेरणा प्राप्त करता है। Xin-She Yang [20] ने निम्नलिखित मान्यताओं के आधार पर जुगुनू एल्गोरिथ्म तैयार किया।

जुगुनू एकनिष्ठ हैं, अर्थात, कोई भी जुगुनू किसी अन्य जुगुनू की ओर आकर्षित हो सकता है।

आकर्षण सापेक्ष चमक के समानुपाती होता है, यानी, एक जुगुनू केवल उन जुगुनुओं की ओर आकर्षित हो सकता है, जो इसकी तुलना में उज्ज्वल हैं, और उनकी ओर बढ़ता है।

किसी दिए गए जुगुनू के लिए, यदि कोई अन्य जुगुनू इसके मुकाबले अधिक चमकीला नहीं है तब यह जुगुनू बेतरतीब ढंग से आगे बढ़ेगा।

एल्गोरिथ्म जुगुनुओं की आबादी (P) को बेतरतीब ढंग से प्रारंभ करने से शुरू होता है। दो जुगुनुओं के बीच की दूरी की गणना करने का सूत्र समीकरण-1 [21] द्वारा दिया गया है।

$$r_{ij}^{(p)} = \| x_i^{(p)} - x_j^{(p)} \| = \sqrt{\Sigma^{d}_{k=1}} \ (x_{i,k}^{(p)} - x_{j,k}^{(p)}) \quad (1)$$

दो जुगुनुओं के बीच के आकर्षण की गणना करने का सूत्र समीकरण –2 द्वारा दिया गया है।

$$B_{ij}^{(p)} (r_{ij}^{(p)}) = (1 - \beta_{min}) \, e^{-\gamma r_{ij}(p)^2} + \beta_{min} \quad (2)$$

यहाँ, βmin न्यूनतम आकर्षण है, और y प्रकाश अवशोषण कारक है।

जुगुनू 'प' जिस गति से जुगुनू 'र' की ओर आकर्षित होती है, उसे निम्नलिखित समीकरण द्वारा दर्शाया जा सकता है।

$$x_i^{(p+1)} = x_i^{(p)} + \beta_{ij}^{(p)} (x_j^{(p)} - x_i^{(p)}) + \zeta k_i^{(p)} \quad (3)$$

यहां, ζ रैंडमाइजेशन पैरामीटर है।

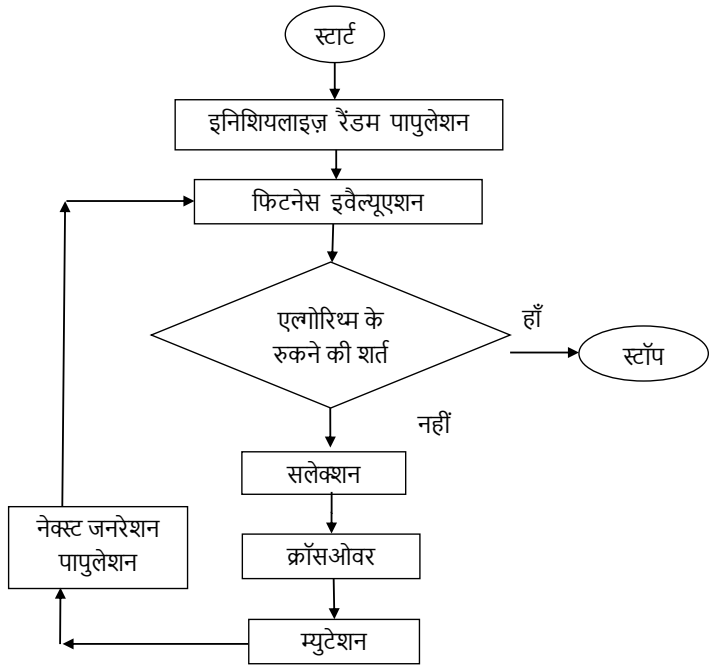
2.2. जेनेटिक एल्गोरिथ्म

जेनेटिक एल्गोरिथ्म विभिन्न अनुकूलन समस्याओं को हल करने के लिए नेचुरल इवोलुशन के कामकाज की नकल करता है। जेनेटिक एल्गोरिथ्म की पुनरावृत्ति में शामिल मूल ऑपरेटरः चयन, क्रॉसओवर और म्यूटेशन हैं। एल्गोरिथ्म की शुरुआत में, जनसंख्या को यादृच्छिक रूप से प्रारंभ किया जाता है, और फिर फिटनेस का मूल्यांकन किया जाता है। इसके बाद, यह सत्यापित किया जाता है कि क्या एल्गोरिथ्म के रुकने की शर्त संतुष्ट है? यदि यह संतुष्ट है, तो प्रक्रिया समाप्त हो गई हैय यदि नहीं, तो अगली पीढ़ी की आबादी प्राप्त करने के लिए जेनेटिक ऑपरेटरों को एक के बाद एक लागू किया जाता है। जब तक कि रुकने की शर्त संतुष्ट नहीं होती, तब तक प्रक्रिया दोहराई जाती है । पारंपरिक जेनेटिक एल्गोरिथ्म का कार्य चित्र–1 द्वारा दिखाया गया है।

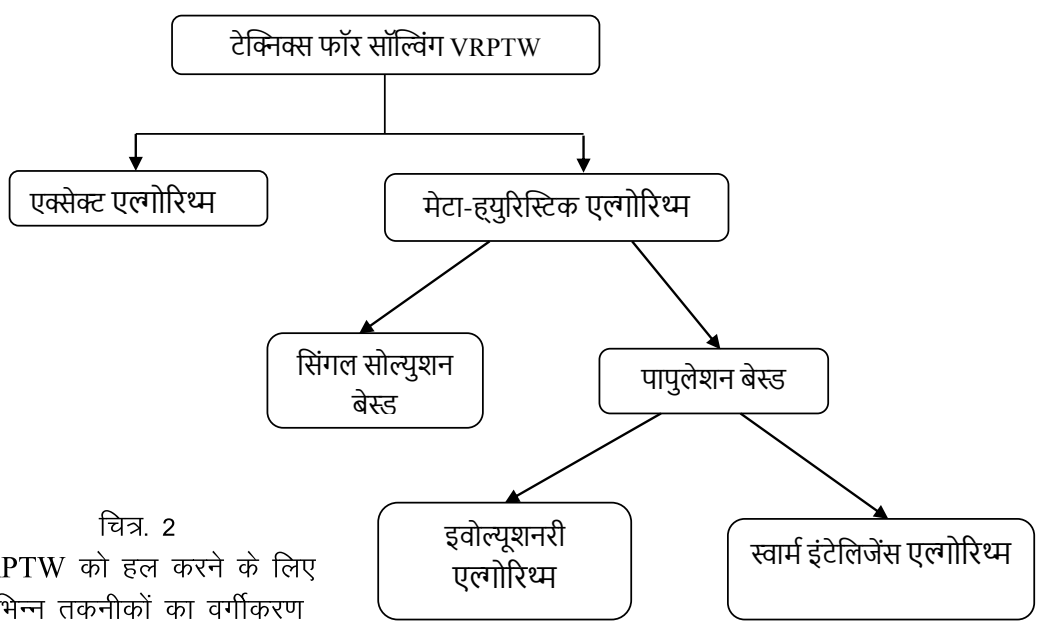
सीमित समय में वाहन रूटिंग समस्या (VRPTW)

VRPTW परिवहन और रसद प्रणालियों में एक महत्वपूर्ण भूमिका निभाता है। यह कई ग्राहकों की सेवा में कई वाहनों के लिए मार्गों के डिजाइन से संबंधित है। VRPTW का लक्ष्य कुल यात्रा दूरी (DT) और वाहनों की संख्या (NV) को कम करना है। VRPTW को हल करने के लिए विभिन्न तकनीकों का एक वर्गीकरण चित्र. 2 में दिखाया गया है।

एक्सेक्ट (Exact) एल्गोरिथ्म हमेशा ही समस्या का इष्टतम समाधान प्रदान करते हैं परन्तु जटिल समस्याओं के लिए ये अनिश्चितकालीन समय भी ले सकते हैं। मेटा–ह्युरिस्टिक एल्गोरिथ्म जटिल समस्याओं का भी निकट इष्टतम समाधान निश्चित समय में प्रदान करने कि क्षमता रखते हैं। सिंगल सोल्युशन बेस्ड मेटा–ह्युरिस्टिक एल्गोरिथ्म एक ही समाधान को समय के साथ बेहतर बनाने का प्रयास करते हैं। पापुलेशन बेस्ड मेटा–ह्युरिस्टिक एल्गोरिथ्म कई समाधानों को समय के साथ बेहतर बनाते हैं तथा उनमे से श्रेष्ठम समाधान प्रस्तुत करते हैं। इवोल्यूशनरी एल्गोरिथ्म प्राकृतिक इवोलुशन प्रक्रिया की नकल करते हैं तथा समय के साथ कई समाधानों को बेहतर करते हैं। स्वार्म इंटेलिजेंस एल्गोरिथ्म प्रकृति में उपस्थित पशु पक्षियों द्वारा झुण्ड में रहकर अपना



चित्र. 1 पारंपरिक जेनेटिक एल्गोरिथ्म का कार्य [12]



चित्र. 2
VRPTW को हल करने के लिए विभिन्न तकनीकों का वर्गीकरण

कार्य सिद्ध करने की कला से प्रेरित होते हैं यह पशु पक्षियों के इस व्यवहार की नकल कर समस्या के कई समाधानों को समय के साथ बेहतर बनाते हैं।

3. सीमित समय में वाहन रूटिंग समस्या का जुगुनू एल्गोरिथ्म द्वारा समाधान

इस शोध पत्र में हमने साहित्य [2] से जुगुनुओं के लिए कण (particle) एन्कोडिंग स्कीमा को अपनाया है। L ग्राहकों और K वाहनों वाले VRPTW में जुगुनू एल्गोरिथ्म को लागू करते समय, प्रत्येक जुगुनू में एक (L + K-1) आयामी वेक्टर (सदिश) होगा। सदिश (1,2, ..., L), (K-1) में अतिरिक्त वर्चुअल ग्राहक जोड़े जाते हैं, जिन्हें मान (L + 1, L + 2, ..., L + k-1) दिया जाता है। इस एन्कोडिंग स्कीमा के अनुसार, (K-1) वर्चुअल ग्राहक जुगुनुओं की स्थिति सदिश को ज्ञ भागों में विभाजित करेंगे। प्रत्येक भाग, उसी क्रम में, एक वाहन को सौंपा गया मार्ग माना जाएगा। यदि केंद्रीय गोदाम को नंबर 0 के साथ सौंपा गया है, तो प्रत्येक मार्ग के लिए, वाहन 0 से शुरू होगा, निर्धारित मार्ग को कवर करेगा, और 0 पर वापस आ जाएगा।

अधिक स्पष्टता के लिए, हम आठ ग्राहकों (L = 8) और तीन वाहनों (K = 3) का उदाहरण लेते हैं। यहां, ग्राहकों को 1,2,3,4,5,6,7,8 और (K-1) के रूप में, दो आभासी ग्राहकों को 9,10 के रूप में सौंपा जाएगा। इसलिए प्रत्येक जुगुनू में सदिश {1,2,3,4,5,6,7,8,9,10} का यादृच्छिक क्रमांकन होगा। बता दें कि परमुटेशन {3,7,9,4,1,5,10,8,2,6} है। चूंकि 9 और 10 आभासी ग्राहक हैं, सदिश को तीन मार्गों में विभाजित किया जाएगा जो तीन वाहनों को सौंपा जाएगा।

उपरोक्त विवरण के अनुसार, इन तीन वाहनों को दिए गए मार्ग निम्नानुसार होंगे:

पहला वाहन: $0 \rightarrow 3 \rightarrow 7 \rightarrow 0$

दूसरा वाहन: $0 \rightarrow 4 \rightarrow 1 \rightarrow 5 \rightarrow 0$

तीसरा वाहन: $0 \rightarrow 8 \rightarrow 2 \rightarrow 6 \rightarrow 0$

जुगुनू एल्गोरिथ्म को VRPTW में लागू करने के चरण निम्नानुसार हैं।

चरण 1: पैरामीटर परिभाषा

मान लीजिए 'द' जुगुनुओं की संख्या हो, 'β_0' आकर्षण की अधिकतम डिग्री हो, 'y' प्रकाश अवशोषण कारक हो, α कदम कारक हो, और maxIter अधिकतम पुनरावृत्तियों की संख्या हो।

चरण 2: यादृच्छिक जनसंख्या की शुरुआत

प्रत्येक जुगुनू के लिए, हम L+K-1 का एक सदिश बनाते हैं, जिसमें 1 से L + K-1 तक संख्याओं का क्रमचय होगा। इस क्रमचय के लिए फिटनेस की गणना करें (VRPTW को हल करें)।

चरण 3: सर्वश्रेष्ठ जुगुनू ढूँढना

समीकरण–2 से आकर्षण की गणना की जा सकती है। फिर, आकर्षण और तीव्रता मूल्य के आधार पर, सबसे चमकदार जुगुनू पाया जा सकता है।

चरण 4: जुगुनुओं की स्थिति में बदलाव

सबसे चमकदार जुगुनू खोजने के बाद, प्रत्येक जुगुनू की स्थिति समीकरण–3 द्वारा अद्यतन की जाती है।

चरण 5: समाप्ति मानदंड

जब समाप्ति मानदंड पूरा हो जाता है, तो वैश्विक सर्वश्रेष्ठ जुगुनू के पास सबसे अच्छा मार्ग होगा और सभी अनुरोधों को पूरा करने के लिए न्यूनतम लागत होगी।

4. आनुवंशिक रूप से ट्यून्ड जुगुनू एल्गोरिथ्म

चूंकि जेनेटिक एल्गोरिथ्म वैज्ञानिक रूप से प्रसिद्ध ऑप्टिमाइजर में से एक है, इसलिए हमने जुगुनू एल्गोरिथ्म के एक पैरामीटर (y) को ट्यून करने के लिए इस शोध पत्र में इसकी क्षमता का उपयोग किया है। प्रस्तावित हाइब्रिड एल्गोरिथ्म को

GFA कहा जाता है। जेनेटिक रूप से ट्यून किए गए जुगुनू एल्गोरिथ्म के कामकाज में शामिल कदम इस प्रकार हैं।

चरण 1: पैरामीटर प्रारंभ

मान लीजिए जनसंख्या का आकार 'n' हो, और 'maxGen' पीढ़ियों की अधिकतम संख्या हो। क्रॉसओवर की दर 0.7 के रूप में सेट की गई है, और म्युटेशन दर 0.01 है।

चरण 2: जनसंख्या प्रतिनिधित्व

जुगुनू एल्गोरिथ्म का पैरामीटर 'y', 0.00 और 1.00 के बीच भिन्न हो सकता है, इसलिए प्रत्येक क्रोमोजोम एक बाइनरी स्ट्रिंग द्वारा दर्शाया जाता है, और 0 और 100 के बीच यादृच्छिक रूप से एक मूल्य प्राप्त करता है।

इस अवधारणा को स्पष्ट करने के लिए, हमने इसे एक उदाहरण के साथ समझाया है।

उदाहरण

व्यक्तिगत प्रतिनिधित्वः 1011100

दशमलव प्रतिनिधित्वः 92

FA का पैरामीटर y: 0.92

चरण 3: फिटनेस कम्प्यूटेशन

अब, जेनेटिक एल्गोरिथ्म की जनसंख्या में प्रत्येक व्यक्ति के लिए, y के मूल्य की गणना उपरोक्त उदाहरण द्वारा बताई गई प्रक्रिया का उपयोग करके की जाती है, और खंड–3 में उल्लिखित चरणों का उपयोग करके VRPTW को हल करने के लिए जुगुनू एल्गोरिथ्म में इसका उपयोग किया जाता है।

चरण 4: क्रॉसओवर

प्रत्येक व्यक्ति के लिए फिटनेस की गणना के बाद, दो पेरेंट्स (P1 और P2) को टूर्नामेंट के चयन का उपयोग करके चुना जाता है। दो संतानों (O1 और O2) को बेतरतीब ढंग से क्रॉसओवर बिंदु का चयन करके उत्पन्न किया जाता है।

उदाहरण के लिए:

P1	: 0 0 0 | 1 0 1 1	(11)
P2	: 0 0 1 | 0 0 1 0	(18)
O1	: 0 0 0 | 0 0 1 0	(02)
O2	: 0 0 1 | 1 0 1 1	(27)

चरण 5: म्युटेशन

इस चरण में, नई उत्पन्न संतानों का एक यादृच्छिक बिट चुना जाता है और स्विच किया जाता है। उदाहरण के लिए:

O1: 0 0 0 0 0 1 (0) (02)

म्यूटेटेड O1: 0 0 0 0 0 1 (1) (03)

इस उदाहरण में, कोष्ठक के भीतर संलग्न बिट म्यूटेटेड है।

चरण 6: समाप्ति मानदंड

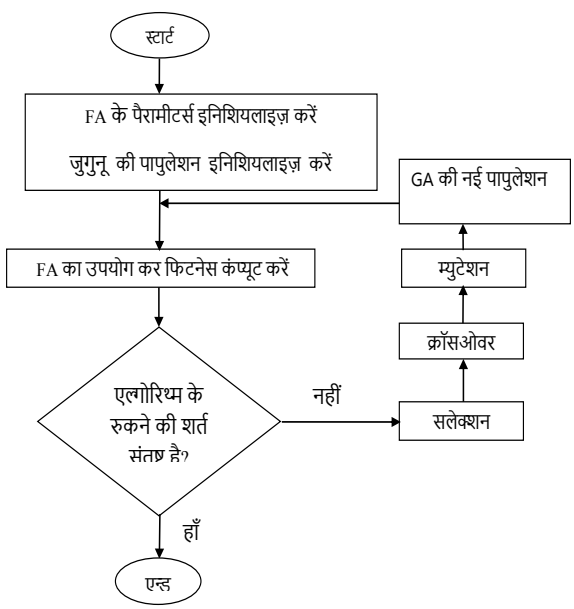
जब समाप्ति मानदंड पूरे हो जाते हैं (maxGen संख्या में पुनरावृत्तियां पूर्ण होती हैं, या कोई अन्य मानदंड पूरे होते हैं), तो अंतिम पीढ़ी में वे पैरामीटर होंगे, जिन्होंने VRPTW के लिए सबसे अच्छा समाधान प्राप्त किया है। ट्यून किए गए फायर फ्लाई एल्गोरिथ्म की मूल प्रक्रिया को एक फ्लोचार्ट के रूप में संक्षेपित किया गया है जैसा कि चित्र. 3 में दिखाया गया है।

5. डेटासेट और सिमुलेशन परिणाम का विवरण

5.1. डेटासेट विवरण

प्रयोग के लिए हमने सोलोमन की 56 बेंचमार्क समस्याओं का उपयोग किया है [14]। इसमें 56 डेटासेट शामिल हैं जो साहित्य में बेंचमार्क के रूप में व्यापक रूप से उपयोग किए गए हैं। संपूर्ण डेटासेट को समस्याओं के छह सेट्स में वर्गीकृत किया गया है। समय विंडो प्रत्येक ग्राहक को आवंटित की जाती है, जिसके भीतर ग्राहक की सेवा करनी होती है।

इस डेटासेट में ग्राहक विवरण का प्रतिनिधित्व, ग्राहक संख्या, X–समन्वय, Y–समन्वय, माल की मांग, समय विंडो का प्रारंभ समय और अंत समय, और सेवा अवधि करते हैं। सोलोमन के डेटासेट में निम्नलिखित छह श्रेणियां हैं: R1, R2, C1, C2, RC1 और RC2। क्लास 'C' निर्दिष्ट करता है कि ग्राहकों को क्लस्टर किया गया है। क्लस्टरिंग या तो भौगोलिक स्थिति या समय विंडो के आधार पर हो सकती है। श्रेणी R में ग्राहकों का भौगोलिक वितरण समान है। RC में, R और C दोनों की मिश्रित विशेषताएं हैं। C1, R1 और RC1 के लिए समय विंडो शेष श्रेणियों की तुलना में संकीर्ण है।



चित्र. 3 प्रस्तावित GFA का फ्लोचार्ट

5.2. सिमुलेशन के परिणाम

हमने सोलोमन के 25, 50 और 100 ग्राहक समस्या उदाहरणों पर प्रयोग किया है। y का मान 0.90 से 1.0 के बीच एक सीमा में अभिसरित हुआ है। सभी गणना इंटेल कोर प5, 4 जीबी रैम के साथ 1.80 गीगाहर्ट्ज प्रोसेसर पर की गई हैं। नीचे दिए गए परिणाम प्राप्त करने के लिए FA को 200 पुनरावृत्तियों के लिए चलाया जाता है। 25 ग्राहक डेटासेट, 50 ग्राहक डेटासेट, और 100 ग्राहक डेटासेट पर प्रयोग के परिणाम क्रमशः तालिका 1, तालिका 2 और तालिका 3 द्वारा दर्शाए गए हैं। हमने 25 ग्राहक डेटासेट के 20 उदाहरण, 50 ग्राहक डेटासेट के 18 उदाहरण और 100 ग्राहक डेटासेट के 11 उदाहरणों को यादृच्छिक रूप से माना है।

तालिका 1: सोलोमन के 25 ग्राहक उदाहरणों के लिए तुलना

S. No.	Problem	Best Known		GFA	
		NV	TD	NV	TD
1	R101	8	617.1	8	618.3
2	R102	7	547.1	7	548.1
3	R103	5	454.6	5	454.7
4	R104	4	416.9	4	418.1
5	R105	6	530.5	6	531.5
6	C101	3	191.3	3	191.8
7	C102	3	190.3	3	190.6
8	C103	3	190.3	3	190.7
9	C104	3	186.9	3	192.1
10	C105	3	191.3	3	191.8
11	C201	2	214.7	2	215.5
12	C202	2	214.7	2	215.4
13	C203	2	214.7	2	215.5
14	C204	2	213.1	2	213.9
15	C205	2	214.7	2	215.5
16	RC101	4	461.1	4	462.2
17	RC102	3	351.8	3	352.7
18	RC103	3	332.8	3	333.9
19	RC104	3	306.6	3	307.1
20	RC105	4	411.3	4	412.4

तालिका 2: सोलोमन के 50 ग्राहक उदाहरणों के लिए तुलना

S. No	Prob-lem	Best Known		GFA	
		NV	TD	NV	TD
1	R101	12	1044	12	1060.12
2	R104	6	625.4	6	638.83
3	R106	5	793	7	865.94
4	R108	6	617.7	6	624.29
5	R110	7	697.0	7	720.40
6	C101	5	362.4	5	363.25
7	C102	5	361.4	5	362.17
8	C103	5	361.4	5	362.17
9	C104	5	358	5	362.88
10	C105	5	362.4	5	363.25
11	C204	2	350.1	2	356.77
12	C208	2	350.5	2	353.56
13	RC101	8	944	8	946.67
14	RC102	7	822.5	7	823.98
15	RC103	6	710.9	6	712.92
16	RC104	5	545.8	5	546.51
17	RC105	4	855.3	4	856.98
18	RC108	3	598.1	3	599.18

तालिका 3: सोलोमन के 100 ग्राहक उदाहरणों के लिए तुलना

S. No	Prob-lem	Best Known		GFA	
		NV	TD	NV	TD
1	C101	10	828.94	10	828.94
2	C106	10	828.94	10	828.94
3	C203	3	591.17	3	591.31
4	C206	3	588.49	3	590.12
5	R101	19	1645.79	19	1694.21
6	R104	9	1007.24	9	1109.32
7	R205	3	994.42	3	1032.68
8	R208	2	726.75	2	745.65
9	RC101	14	1696.94	15	1786.32
10	RC106	11	1424.73	11	1496.39
11	RC205	4	1297.19	4	1352.25

परिणामों की व्याख्या

सभी तालिकाओं में, TD कवर की गई कुल दूरी का प्रतिनिधित्व करता है, और NV उपयोग किए गए वाहनों की संख्या का प्रतिनिधित्व करता है। VRPTW में TD और NV के छोटे मूल्य वांछनीय हैं।

यह तालिका 1 से देखा जा सकता है कि GFA के लिए NV का मान सभी 20 उदाहरणों के लिए सबसे प्रसिद्ध मूल्य के बराबर है (पंक्ति 1 में यह '8'है, पंक्ति 2 में '7' है, और इसी तरह पंक्ति 20 में यह '4' है)। और GFA के लिए TD का मूल्य सभी 20 उदाहरणों के लिए सबसे बेहतरीन ज्ञात मूल्य के करीब है।

तालिका 2 से यह देखा जा सकता है कि GFA के लिए NV का मान 18 में से 17 उदाहरणों के लिए सबसे प्रसिद्ध मूल्य के बराबर है, पंक्ति 3 में 'R106' को छोड़कर, जहां सबसे प्रसिद्ध NV '5' है, और GFA के माध्यम से प्राप्त NV '7.' है। जेनेटिक रूप से ट्यून किए गए जुगुनू एल्गोरिथ्म के लिए TD का मान 18 में से 16 उदाहरणों के लिए सबसे प्रसिद्ध मूल्य के बहुत करीब है।

तालिका 3 से यह देखा जा सकता है कि GFA के लिए NV का मान 11 उदाहरणों में से 10 के लिए सर्वोत्तम ज्ञात मूल्य के बराबर है, पंक्ति 9 में 'RC101' को छोड़कर, जहां सबसे प्रसिद्ध NV '14 है, और GFA के माध्यम से प्राप्त NV '15 है। जेनेटिक रूप से ट्यून किए गए जुगुनू एल्गोरिथ्म के लिए TD का मूल्य 11 में से 6 उदाहरणों (पंक्ति 1, पंक्ति 2, पंक्ति 3, पंक्ति 4, पंक्ति 6, और पंक्ति 8) में सर्वोत्तम ज्ञात मूल्य के करीब है और शेष 5 उदाहरणों (पंक्ति 5, पंक्ति 7, पंक्ति 9, पंक्ति 10, और पंक्ति 11) में भी यथोचित करीब है।

परिणामों से स्पष्ट है कि GFA 25 ग्राहक समस्या के साथ–साथ 50 ग्राहक समस्या के सभी

मामलों में लगभग इष्टतम परिणाम प्राप्त करता है। अधिक जटिल 100 ग्राहक समस्या के लिए भी, GFA ने निकट–इष्टतम समाधान प्राप्त किए हैं, इस प्रकार VRPTW को हल करने के लिए अपनी मजबूती और दक्षता स्थापित की है।

निष्कर्ष और भविष्य में विस्तार की संभावनाएं

इस शोध पत्र में हमने FA का नया संस्करण जेनेटिक रूप से ट्यून किए गए जुगुनू एल्गोरिथ्म पेश किया है । इस दृष्टिकोण में जुगुनू एल्गोरिथ्म के एक पैरामीटर (y) को बेहतर परिणाम प्राप्त करने के लिए जेनेटिक एल्गोरिथ्म द्वारा ट्यून किया गया है। VRPTW के लिए सोलोमन के बेंचमार्क डेटासेट के विभिन्न उदाहरणों पर जेनेटिक रूप से ट्यून किए गए जुगुनू एल्गोरिथ्म लागू किया गया है, जिसमें 25 ग्राहक, 50 ग्राहक और 100 ग्राहक शामिल हैं। तालिका–1 और तालिका–2 द्वारा दर्शाए गए परिणामों से यह स्पष्ट है कि प्रस्तावित एल्गोरिथ्म ने समाधान दिए हैं जो 25 ग्राहकों और 50 ग्राहकों वाले डेटासेट के लिए संबंधित सबसे प्रसिद्ध समाधानों के बहुत करीब हैं। यहां तक कि 100 ग्राहकों वाले बड़े डेटासेट के लिए भी जेनेटिक रूप से ट्यून किए गए जुगुनू एल्गोरिथ्म ने इष्टतम मूल्य के यथोचित करीब समाधान प्राप्त किए हैं, जैसा कि तालिका–3 द्वारा दर्शाया गया है। हम यह निष्कर्ष निकालते हैं कि प्रस्तावित जेनेटिक रूप से ट्यून किए गए जुगुनू एल्गोरिथ्म में VRPTW को हल करने की क्षमता है।

इस शोध पत्र में हमने फायर फ्लाई एल्गोरिथ्म के केवल एक पैरामीटर को ट्यून किया है। प्रस्तावित दृष्टिकोण जुगुनू एल्गोरिथ्म के एक से अधिक पैरामीटर को फाइन–ट्यून कर सकता है। इसके अलावा, प्रस्तावित दृष्टिकोण को कई अन्य जटिल इष्टतमीकरण समस्याओं पर भी लागू किया जा सकता है।

तालिका 4: अंग्रेजी भाषा के तकनीकी शब्दों के समकक्ष उपयोग किये गए हिंदी शब्दों की सूची

Firefly Algorithm	जुगुनू एल्गोरिथ्म
Light absorption factor	प्रकाश अवशोषण कारक
Nature-Inspired algorithms	प्रकृति–प्रेरित एल्गोरिथ्म
Optimal	इष्टतम
Optimization	इष्टतमीकरण
Vehicle Routing Problem	वाहन रूटिंग समस्या
Vehicle Routing Problem with Time Windows	सीमित समय में वाहन रूटिंग समस्या

References :

[1] Yang, X. S. Firefly Algorithms for multimodal optimization. International symposium on stochastic algorithms, Springer Berlin Heidelberg, 2008; 169-178.

[2] Wu, Y., Ye, C. M., Ma, H. M., & Xia, M. Y. Parallel particle swarm optimization algorithm for vehicle routing problems with time windows. Computer Engineering and Applications. 2007; 43(14): 223-226.

[3] Abualhaija, S., & Zimmermann, K. H. (2016). D-Bees: A novel method inspired by bee colony optimization for solving word sense disambiguation. *Swarm* and *Evolutionary Computation*, *27*, 188-195.

[4] Tilahun, S.L., Ong, H.C. Modified Firefly Algorithm. Journal of Applied Mathematics doi:http://dx.doi.org/10.1155/ 2012/467631, 2012.

[5] Chandrasekaran, K., and Sishaj P. Simon. "Network and reliability constrained unit commitment problem using binary real coded firefly algorithm." *International Journal of Electrical Power & Energy Systems* 43.1 (2012): 921-932.

[6] Krishnanand, K. N., and Debasish Ghose. "Detection of multiple source locations using a glowworm metaphor with applications to collective robotics." *Swarm intelligence symposium, 2005. SIS 2005. Proceedings 2005 IEEE*. IEEE, 2005.

[7] Damodaram, D., and T. Venkateswarlu. "FPGA implementation of genetic algorithm to detect optimal user by cooperative spectrum sensing." ICT Express. 2017.

[8] Sohn, Insoo. "New SLM scheme to reduce the PAPR of OFDM signals using a genetic algorithm." ICT Express 2.2. 2016: 63-66.

[9] Li, J., Huang, Y., & Niu, X. A Branch population Genetic-Algorithm for Dual-Resource Constrained Job Shop Scheduling Problem. Comput Ind Eng. 2016; 102: 113-13. https://doi.org/10.1016/j.cie.2016.10.012

[10] Lim, T. Y., Al-Betar, M. A., & Khader, A. T. Taming the 0/1 knapsack problem with monogamous pairs Genetic-Algorithm. Expert Syst Appl. 2016; 54: 241–250. https://doi.org/10.1016/j.eswa.2016.01.055.

[11] Mishra, A., & Shukla, A. (2016). Mathematical analysis of the cumulative effect of novel ternary crossover operator and mutation on probability of survival of a schema. *Theoretical Computer Science*, 1–11. http://doi.org/10.1016/j.tcs.2016.07.035

[12] Mishra, A., & Shukla, A. (2017). Mathematical analysis of schema survival for genetic algorithms having dual mutation. *Soft Computing*, 1-9. https://doi.org/10.1007/s00500-017-2650-3

[13] Mishra, A., & Shukla, A. (2018). A new insight into the schema survival after crossover and mutation for genetic algorithms having distributed population set. *International Journal of Information Technology*, *10*(2), 165-168. https://doi.org/10.1007/s41870-018-0087-2

[14] Dantzig, G. B., Ramser, J. H. The truck dispatching problem, Manage Sci. 1959; 6(1): 80-91.

[15] Toth, P., Vigo, D. The vehicle routing problem. Philadelphia, PA: SIAM Monographs on Discrete Mathematics and Applications, Society for Industrial and Applied Mathematics. 2001.

[16] Goel, R., & Maini, R. (2018). A hybrid of ant colony and firefly algorithms (HAFA) for solving vehicle routing problems. *Journal of Computational Science*, *25*, 28-37.

[17] Osaba, Eneko, et al. "An evolutionary discrete firefly algorithm with novel operators for solving the vehicle routing problem with time windows." *Nature-Inspired Computation in Engineering*. Springer, Cham, 2016. 21-41.

[18] Taha, Anass, Mohamed Hachimi, and Ali Moudden. "A discrete Bat Algorithm for the vehicle routing problem with time windows." *Logistics* and *Supply Chain Management (LOGISTIQUA), 2017 International Colloquium on*. IEEE, 2017.

[19] Dixit A., Mishra A., Shukla A. (2019) Vehicle Routing Problem with Time Windows Using Meta-Heuristic Algorithms: A Survey. In: Yadav N., Yadav A., Bansal J., Deep K., Kim J. (eds) Harmony Search and Nature Inspired Optimization Algorithms. Advances in Intelligent Systems and Computing, vol 741. Springer, Singapore. https://doi.org/10.1007/978-981-13-0761-4_52

[20] Yang, X. S., & He, X. (2013). Firefly algorithm: recent advances and applications. *International Journal of Swarm Intelligence*, *1*(1), 36-50.

[21] Casciati, Sara, and Lorenzo Elia. "The potential of the firefly algorithm for damage localization and stiffness identification." *Recent Advances in Swarm Intelligence* and *Evolutionary Computation*. Springer, Cham, 2015. 163-178.