



A solution to dynamic green vehicle routing problems with time windows using spiking neural P systems with modified rules and learning

Resmi Ramachandranpillai¹ · Michael Arock¹

Accepted: 13 January 2021 / Published online: 15 February 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC part of Springer Nature 2021

Abstract

Dynamic routing problems involve electronic decision making, which compromises reactivity with the quality of decision making. The time for seeking better decisions comes at the price of a lower reactivity to changes in inputs. This factor is especially important in situations where consumers are calling for a service and a good decision needs to be made as quickly as possible. There are two approaches in solving dynamic vehicle routing (DVRP): to run a state solver every time new requests come in, to construct an initial solution and then to update it each time new information comes. The latter is more commonly used and more flexible. Also, the environment is badly affected by factors like CO₂ emissions; noise; etc. A variant of VRP called green VRP (GVRP) has been formulated in this context. The solution strategies for GVRP are developed to help organizations with alternative fuel-powered vehicles to resolve challenges that arise in conjunction with limited refueling facilities as a result of restricted vehicle driving range. Here, the authors propose a spiking neural P system (SN P)-based model with modified rules and learning in association with firefly optimization (FA) to solve the combined version of GVRP and D VRP with time windows, called DGVRPTW. The SN P system proposed here is a multilayer neural system with embedded potentials and learning facilities which uses the rectified linear unit (reLu) as activation functions. The proposed SN P system is used for geo-location clustering, and the firefly algorithm (FA) is used for route optimization. The proposed SN P system can do predictions accurately when a new customer enters the scenario. The scheme has been tested on medium as well as large-scale instances and analyzed different performance measures such as nature of convergence, utilization rate, solution improvement percentage and dynamic measures. Having applications in image classifications, optimization problems, etc., the proposed system is worthy of future study.

Keywords Spiking neural P systems · Dynamic green vehicle routing problems · Rectified linear unit · Firefly algorithm

1 Introduction

Natural computing (NC) [1] is a term used to describe those methods inspired by nature to create innovative problem-solving techniques or those focused on the use of computer systems to imitate natural phenomena or those which use bio-inspired materials for processing. The main research fields that involve NC are artificial neural networks (ANN) [2], swarm intelligence [3], evolutionary algorithms [4], quantum computing [5] and membrane computing (MC) [6]. It might seem that the relationship between computer science and biology is boosting at the moment, with positive impacts for both fields. Biology is a rich source of ideas for creating new solutions and paradigms for computationally hard problems, generating new concepts and hypotheses that help to describe and evaluate different systems. By the end of 1990s, membrane computing (MC) had emerged as a branch of natural computing. The models pertaining to this computational framework, called P systems, were considered as distributed devices inspired by cell structure. Some of these models were also taken off the structure and function of more complex entities such as tissues and organs. MC began as a new experimental paradigm that created a wide range of basic research in theoretical computer science. Membrane systems have been used in recent years as modeling vehicles for different biological systems or as design languages in graphics or to represent a wide variety of parallel systems. Because of its non-determinism and maximum parallelism features, many engineering design optimization problems are solved with MC at a reasonable time [7]. Membrane structure classifies MC systems into cell-like [8], tissue-like [9], and neural-like [10] structures. The latter group is MC's new branch, which is integrated into the P systems (SN P systems) by spiking neurons.

Spiking neural P systems (SN P systems) are bio-inspired neural computation models that are designed by abstracting the way the neurons spike. This means the coordination of biological neurons is accomplished by spikes in central nervous systems. SN P systems come under artificial neural network models of the third generation.

An SN P system can be represented by a direct graph with nodes referring to spiking neurons and edges shape synapses between neurons that can represent SN P systems. SN P systems with the division of neurons [11], budding [12] and spikes and anti-spikes [13] are implemented in many applications and have proven to be efficient and reliable. SN P systems have been viewed as word generative devices of 0's and 1's [14]. To solve knapsack problems, an optimization SN P system was introduced to find solutions to the unconstrained single objective optimization problem [15]. It was applied to solve knapsack instances and proved to be efficient. Also, the use of SN P systems can be seen in many smart and expert fields, such as the semantics of deductive database systems [16] and multiples in parallel [17]. It is therefore concluded that SN P systems are more powerful in both theoretical and practical aspects of intelligent and expert systems [18].

The Vehicle Routing Problem (VRP) [19] is the problem of constructing suitable routes from a depot to a range of destinations each with business-specific

restrictions, such as vehicle limits, cost controls, time periods and resources limitations related to the depot loading process. If all the services start and end in some fixed time interval, then it is called VRP with time windows (VRPTW) [20]. VRP is known for its computational complexity for which many exact and heuristic algorithms have been proposed, but arriving optimal solutions is still a challenging task. Researchers almost define some basic information about the locations and demands of customers, available vehicles, and so on. In most VRP studies, these are fully known before the service plan is carried out. However, VRP is dynamic (DVRP) [21] in actual service processes; that is, the demands and arrangements of customers are gradually changing over time, although some of the demands of customers may be known beforehand. Also, since 2014, most of the researchers explore green VRP (GVRP) [22], a new variant of VRP, which focuses on overcoming difficulties of limited refueling infrastructure and minimizing CO₂ emissions.

A meta-heuristic suggested by Yang in 2008 is inspired by nature is based on fireflies' flashing actions, which serves as a signal system for attracting other fireflies. As can be seen in several surveys [23–25], since its proposal, the FA has been applied successfully to many different fields of optimization problems. Nevertheless, in the current scientific community, it still attracts a lot of attention. Since FA has been designed for continuous optimization problems, there are only very few applications of FA in VRP instances. This lack of work, together with the minimal computational overhead and the good performance that the FA has demonstrated since its proposal, has motivated its use in this study.

The aim of this paper is to develop an efficient spiking neural P system with modified rules, potentials and learning capabilities to solve the combination of all of the above-mentioned variants, called dynamic green VRP with time windows (DGVRPTW), which is more complex because of the additional constraints and objectives added. The suggested SN P system here is a multilayer SN P system used for the region prediction of dynamic customers followed by an FA for route optimization.

The main contributions of the proposed work are:

1. A mathematical model of the DGVRPTW has been formulated with some added constraints which optimize the refueling capacity of vehicles and CO₂ emissions.
2. A multilayer SN P system with learning and training has been designed with activation rules on synapses for tackling dynamism.
3. An ideal arrangement of neurons in the SN P system is presented which carries information and does the processing accurately.
4. A new way of encoding fireflies and distance calculation is adopted for optimizing routes inside each cluster.

The remaining sections are methodized as follows. Section 2 carries out a detailed literature survey on the problem. Section 3 talks about classical FA. Section 4 presents details about DGVRPTW, dynamism followed by the mathematical formulation of the problem. Section 5 gives the design of SN P systems with

potentials. Section 6 gives a detailed description of different modules, structure, and functioning of the proposed system. Experimental analysis and results are given in Sect. 7. Finally, the conclusion and future scope of the work are presented in Sect. 8.

2 Literature review

2.1 Dynamic vehicle routing problem: survey

Technology innovation in various fields has fostered the development of intelligent transportation systems (ITS), using the advent of geo-location techniques with geographic information systems [26]. The object of vehicle routing is an administrative function linked to the competence of dispatch as well as the cost of optimization directly dependent on the size of the fleet [27]. The state-of-the-art schemes in ITS use an optimization technique coupled with some heuristic search techniques to attain a solution.

Additional constraints such as regulations on waiting, service and travel times characterize the transportation of people. Cab system is the most common individual online transportation system where the requests from the customers are made up of pickup time and location coupled with the drop location. There are many variants available, such as advance booking, sharing, etc. [28–31].

Owing to the highly variable travel time, factors such as congestion, rivalry, and cooperation between transport companies should be taken into consideration. All these applications fall under the logistics of the region. Effective scheduling of real-time traffic and dynamic routing of a fleet of vehicles require additional modules and attributes. For better trade-off between dynamic travel and service times, a decision support system (DSS) can be used [32–34].

In courier transport systems, not only request locations, time windows and capacity constraints should be taken into account, but also traffic data and prohibited paths should also be considered. The Automatic Fleet Management System (AFMS) enabled optimization has been considered to solve these types of problems [35–37].

Customer satisfaction is an important factor in FMS. Delivery of newspaper is an example of such a type of domains where complaints will be filed in the event of delivery delay. Centralized applications were proposed to reduce costs and improve customer satisfaction, using cell phones to communicate continuously with drivers at the same time as routing [38, 39].

The retailer designs different customers in grocery delivery systems that can be considered within a fixed time span. At the same time, when the capacity constraints are violated, it is made inaccessible to the customers. A customer's time slots are actively designed based on future in-home delivery request issues [40]. Greedy randomized adaptive search procedure (GRASP) and adaptive large neighborhood search (ALNS) have been proposed that consider uncertainty by introducing scenarios for possible realizations of demand. Problems with dynamic stacker cranes [41, 42] which considered container carriers with loading and unloading vessels are

undergoing operational study. However, factories and hospitals are other areas of operation where goods or medical devices have to be moved, respectively [43].

The capacity constraints can be exceeded in the service domain. The dynamic traveling salesman problem [44] is one of the most illustrative models in this domain. Certain fields provide activities for repair and maintenance, winter gritting, etc. A daily visit to customer locations is mandatory in the first groups. A customer can also order for a new service in the event of an emergency. A context is that the French non-profit organization operates under the call of patients with a crew of physicians supported by certain emergency services. A subset of streets or roads that are impacted by the storm must be gritted in winter gritting applications. Depending on the path of the storms, the new sections have to be gritted [45].

2.2 Green vehicle routing problem: survey

Green-VRP (G-VRP) research focuses on optimizing transport energy consumption. The cost of gas accounts for a large part of the total transportation cost of petroleum [46]. The most obvious course of action would be to reduce fuel consumption and improve transportation performance at the operational level. It is also important that a reduction in fuel consumption dependent on petroleum will substantially reduce greenhouse gas emissions [47]. To include fuel consumption in the routing model, it is important to formulate computing fuel consumption in relation to the condition of a moving vehicle. In [48], the authors considered a more practical transport cost impacting both the load of the vehicle and the length of the traveled arc. They describe the Energy Minimizing Vehicle Routing Problem (EMVRP) as the CVRP with a new cost objective, where the cost function is a combination of the total load (including the empty vehicle weight) and the arc duration. A model of fuel consumption is given in [49]. They suggested a Fuel Consumption Rate (FCR) known to be CVRP (FCVRP), which would extend CVRP to reduce fuel consumption. In [50], the authors applied the transport speed to the fuel consumption measurement model in time-dependent VRPs in addition to the transport range and the charging weight discussed in the above two articles. Other VRP-related studies that aim at minimizing total fuel consumption are also been designed [51–54]. This design seeks to eliminate the possibility of running out of fuel with the goal of reducing the total distance. Stegner [55] extended time-windowed G-VRP. The above papers on GVRP state that fuel consumption merely provides the formula for calculating fuel consumption, assuming that the fuel is sufficient to cover the entire tour.

State-of-the-art schemes have only considered the DVRP without any additional constraints such as CO₂ emissions, fuel consumptions etc. Also, it lacks concrete learning and training methods to predict the dynamic regions for customers. Instead they make use of schedulers to convert a dynamic requests to static requests over time periods and then send to the optimization procedures for optimal solutions.

The proposed system is no way similar to the schemes in literature. The authors try to approach the problem from a different perspective. Here, the maximum parallelism feature of SN P systems are coupled with learning and training methods in order to predict the dynamic customers on arrival rather than converting it to static

requests by schedulers. Thus, the proposed system replaces the schedulers and predicts the region of customers based on the trained data. Moreover, it is for the first time, the SN P systems are employed for such specific tasks with learning and training functions in VRP domain, which highlights the novelty of the system.

3 Classical firefly algorithm

CFA was first introduced in [56], which was based on the flashing behavior of fireflies for drawing in prey and mates. The main rules used in FA are:

1. Fireflies will be attracted to each other regardless of their gender.
2. As the separation between the two fireflies' increases, their attraction decreases. For any two fireflies, the less bright one will move towards the brighter one. In the event that the brightness is the same or less, it will move arbitrarily.
3. The landscape of the objective function is dependent on the brightness of the firefly.

The attraction of a firefly depends upon the light intensity and this relation can be expressed by the following formula:

$$\beta(r) = \beta_0 D, \text{ where } D = e^{-\gamma r^2} \quad (1)$$

where γ is the light absorption coefficient.

Also, the distance between two fireflies at positions i and j is calculated by:

$$r_{ij} = \left\| X_i - X_j \right\| \quad (2)$$

The movement of a less bright firefly i towards a brighter one j can be calculated by:

$$X_i^{\text{new}} = X_i^{\text{old}} + \beta(r_{ij})(X_j - X_i) + \alpha(\text{rand} - 0.5) \quad (3)$$

where α a randomization parameter and rand is a random number within the uniform distribution over $[0, 1]$.

4 Problem description and formulation

4.1 Dynamic green vehicle routing problem with time windows

Given a complete graph G with nodes representing client locations, depot and fuel stations, the DGVRPTW finds a set of routes, each of which starts and ends at depot, visit a set of clients within the specified time window without violating the vehicle capacity and fuel tank capacity with minimum total distance by each of the vehicles. Because of the recent improvements in real-world communications,

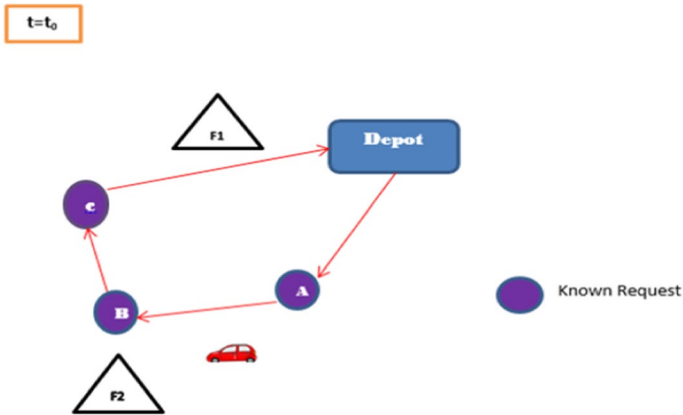


Fig. 1 Static environment route planning (at $t=t_0$)

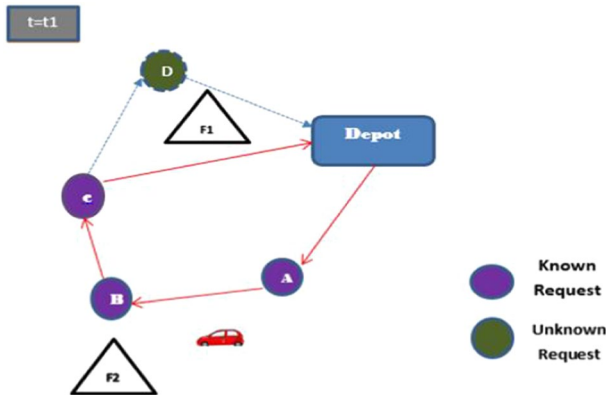


Fig. 2 Dynamic environment route planning (at $t=t_1$)

‘dynamism’ would be very significant. In dynamic routing, not all pieces of information on the routing process are available when it begins, and the information may change even after the routes have been constructed. Figures 1 and 2 show the route planning procedure in the static and dynamic environment, respectively. The dotted lines in Fig. 2 show the new path constructed based on the dynamic request from the customer D. The F1 and F2 are fuel stations.

4.2 Degree of dynamism

The performance of dynamic routing is calculated based on the number of dynamic demands and the time when these demands occur, while static VRP depends on the number and distribution of clients. If we examine the performance of a specific algorithm under constraints, the measure of ‘dynamism’ would be important. The

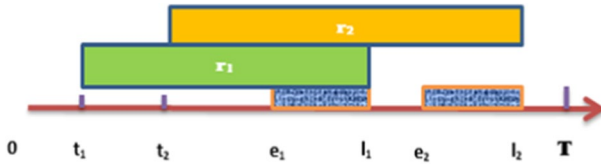


Fig. 3 Reaction time of two customers in dynamic scenarios with a time window

reaction time of 2 customers with time windows under dynamic situation is given in Fig. 3, where r_1 and r_2 are reaction times of customer 1 and customer 2, respectively.

In case of time windows (t_w), the effective dod (e_{dod}) can be defined as:

$$\begin{aligned}
 e_{dod} &= \frac{1}{n} \sum_{k=1}^n \left(\frac{T - (l_k - t_k)}{T} \right) + t_w \\
 &= \frac{1}{n} \sum_{k=1}^n \left(1 - \frac{r_k}{T} \right) + t_w
 \end{aligned}
 \tag{4}$$

where $0 \leq e_{dod} - t_w \leq 1$, $l_k - t_k \leq T, k = 1, 2, ..n$, where n is the total number of requests and the planning horizon is defined in the interval $l[0, T]$.

4.3 The mathematical formulation of the problem

DGVRPTW is defined based on a complete graph $G=(V, E)$, where $V=\{1, 2, M, M+1\}$ denotes the vertex set and $E=\{\{(i, j) \mid i, j \in M, i \neq j\}$ denotes the edge set. Here, $V_{cust} = \{1, 2, \dots, M\}$ is the customer set of vertices. $M+1$ is the depot vertex. The method of delivery of the commodity to the set of customers is carried out using the same set of M vertices with a fixed maximum capacity Q for each vehicle. The depot D is fitted with a collection of n vehicles.

Each vertex $i \in V$ is associated with several measures such as traveling cost/distance denoted as $C_{ij}, (i, j) \in E$ and $i \in M, j \in M$, a request r_i , time of service by vehicle k, t_{ik} and the time window $[e_i, l_i]$, where e_i represents the earliest time and l_i represents the latest time when the service at vertex i can start. Also, the speed function f is defined by p non decreasing levels p^f over set $S = \{1, 2, p\}$. So, the problem is to find the minimum total traveling cost subject to vehicle capacity, time window, and fleet size that visit each customer in the set V_{cust} . But, while servicing the customers, each vehicle emits carbon dioxide (CO2) which is proportional to the total fuel used by vehicles.

So, the main objectives of DGVRPTW are to find routes with minimal total traveling cost, total fuel consumption and CO2 emissions subject to the following conditions:

The vehicle leaves the depot D and returns to D after completion of customer service.

1. The number of vehicles used at the depot cannot exceed the size of the fleet.
2. Each customer is served once by one vehicle.
3. The service to the customers must start with the time window $[e_i, l_i]$
4. The total capacity of each vehicle cannot exceed Q
5. The vehicle's early arrival is allowed, but the vehicle must wait for the service to be delayed to time e_i but not later than l_i
6. Emissions of CO₂ are proportional to the overall vehicle fuel consumption

The mathematical model of DGV_{RPTW} involves a group of decision variables that model the sequence of vehicles visiting customers and the following are defined:

$$y_{ijk} = \begin{cases} 1, & \text{if vehicle } k \text{ travels from } i \text{ to } j \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

$$x_{ijk} = \begin{cases} 1, & \text{if vehicle } k \text{ traverses edge } (i, j) \text{ with speed } f \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

$P_{ijk} \in R+$: the total amount of flow on each edge $(i, j) \in V$.

$Z_i \in R+$: the time which service starts at node $i \in V$.

The mathematical model of the problem can be formulated as:

Minimize.

- The total traveling cost performed by each of the vehicles.

$$\sum_{i \in V} \sum_{j \in V, i \neq j} \sum_{k \in N} C_{ij} y_{ijk} \tag{7}$$

- Minimize the total fuel usage and CO₂ emissions
- The total fuel usage of each of the vehicles is directly proportional to the CO₂ emissions. The fuel rate [57] can be expressed by:

$$\text{Fuel rate} = \xi \left(\omega GV + \frac{P}{\psi} \right) / k, \tag{8}$$

where ξ -fuel to air ratio;

ω - engine friction factor.

G-engine speed.

V-engine displacement.

ψ and k are constants.

P-engine power output per second which is given by:

$$P = P_{\text{total}} / \varphi_{tf} + P_{re}, \text{ where,} \tag{9}$$

P_{re} —engine power needed for running losses of the engine.

φ_{tf} —vehicle drive train efficiency.

P_{total} —total tractive power in kilowatts

$$\begin{aligned}
 & \sum_{i \in V} \sum_{j \in V, i \neq j} \sum_{k \in N} \omega G V \lambda C_{ij} \sum_{f \in S} x_{ijk}^f / r^p + \sum_{i \in V} \sum_{j \in V, i \neq j} \sum_{k \in N} \omega \lambda \alpha_{ij} C_{ij} \delta y_{ijk} + \sum_{i \in V} \sum_{j \in V, i \neq j} \sum_{k \in N} \lambda \alpha_{ij} C_{ij} \delta P_{ijk} \\
 & + \sum_{i \in V_{cust}} \sum_{j \in V_{cust}, i \neq j} \sum_{k \in N} \beta \delta \lambda C_{ij} \sum_{f \in S} x_{ijk}^f (r^f)^2 \delta y_{ijk}
 \end{aligned} \tag{10}$$

subject to:

$$1. \quad \sum_{j \in V_{cust}} \sum_{k \in N} y_{ijk} \leq |N| \tag{11}$$

$$2. \quad \sum_{j \in V} \sum_{k \in N} y_{ijk} = 1, \quad i \in V_{cust} \tag{12}$$

$$3. \quad \sum_{i \in V} \sum_{k \in N} y_{ijk} = 1, \quad j \in V_{cust} \tag{13}$$

$$4. \quad \sum_{d \in V_{M+1}} \sum_{j \in V_{cust}} y_{djk} \leq 1, \quad k \in N \tag{14}$$

$$5. \quad \sum_{d \in V_{M+1}} \sum_{j \in V_{cust}} y_{idk} \leq 1, \quad k \in N \tag{15}$$

$$6. \quad \sum_{i \in V_{M+1}} \sum_{j \in V_{M+1}} y_{ijk} = 0, \quad k \in N \tag{16}$$

$$7. \quad \sum_{j \in V} P_{jik} - \sum_{j \in V} P_{ijk} = t_i, \quad i \in V_{cust}, \quad k \in N \tag{17}$$

$$8. \quad \sum_{f \in S} x_{ijk}^f = y_{ijk}, \quad i, j \in V, \quad k \in N \tag{18}$$

$$9. \quad y_{ijk} \in \{0, 1\}, \quad i, j \in V, \quad k \in N \tag{19}$$

$$10. \quad P_{ijk} \in R^+, \quad i, j \in \{V, k \in N\} \tag{20}$$

$$11. \quad x_{ijk}^f \in \{0, 1\}, \quad i, j \in V, \quad k \in N, \quad f \in S \tag{21}$$

5 Design of spiking neural P systems with potentials

A variation in the SN P system is designed to select rules in a simple manner appropriately as in [58]. A small rule modification of the rule has been done in order to introduce the firing process in the proposed system. To this extent, we do not include spikes as the regular SN P systems, yet it is believed that each neuron contains a potential that can be communicated as a rule. If the potential is less than or equal to its threshold potential, each neuron fires. At that point, the neuron will devour a fraction of the potential and the remaining potential is transmitted to the neighboring neurons by the neurotransmitters, which may increase or decrease with a weight in the positive and negative form, respectively.

$$\Pi = (A, \sigma_1 \dots \sigma_m, Sn, fn, In, Ot)$$

Where

1. A is the alphabet set.
2. $\sigma_1, \dots, \sigma_m$ are neurons of the tuple $\sigma_i = (P_i, \gamma_i, R_i) \ 1 \leq i \leq m$, where:
 - (1) P_i is the potential that is initially given to neuron σ_i
 - (2) γ_i is a real number called output weight of neuron σ_i ;
 - (3) R_i is the rule set in σ_i of the form:
 - $E/a^k \rightarrow a; d$, where E is a regular expression over the alphabet A , $k \geq 1$ and $d \geq 0$. At some point in time, t , if neuron σ_i contains c spikes and $c \in a^k L(E)$, $c \geq k$, then this rule will be activated, consuming k spikes, sending one spike to each of the neighboring neurons after d time steps. At the step $t+d$, the neuron again becomes active and can receive spikes and is able to apply the rule at $t+d+1$.
 - $Th_i/d_j \rightarrow 1, f$, where $j=1, 2, \dots, r_i$, for some $r_i \geq 1$, where $Th_i \in Rc$, and $f \geq 0$, $Th_i \geq 0$ is the potential of neuron σ_i and $d_j \in Rc$ with the constraint $0 < d_j < Th_i$.
3. Sn_t is the synapses at step t of the form (j, k, γ_{jk}) . γ_{jk} is the weight of the synapse connecting the neurons σ_j and σ_k and is used to amplify the signals. If the synapse gets only one spike, then the weight γ_{jk} is multiplied by 1 and sends to the receptor neuron. Generally $Sn \subseteq \{1, 2, \dots, m\} \times \{1, 2, \dots, m\} \times Rc$ are neurotransmitters between the neurons of the system where $i \neq j, k \neq 0$, for $(i, j, k) \in Sn$.
4. $In \in \{\sigma_1, \dots, \sigma_m\}$ is the input set of neurons.
5. $Ot \in \{\sigma_1, \dots, \sigma_m\}$ is the output set of neurons.

The information in the form of spike trains is read by the input neurons from the environment. Based on the spike train configurations it spikes and passes information to the synapses. The output neurons are responsible for emitting spikes to the environment and $In \cap Ot = \emptyset$.
6. Learning function fn is used to amplify or weaken the synapses during the activations.

Note that there are no forgetting rules inside each neuron yet it is controlled by adjusting the neuron's potentials.

6 Proposed system design

State-of-the-art algorithms use either ANN to group customers into different clusters using x and y coordinate positions that represent customers' geo locations or they use any clustering technique to group customers based on locations. In ANN, the neural synapses are weighted by weight vectors. In all those cases, the dynamic requests from customers are converted into static requests by an event scheduler. Density-based models have also been used to identify the nearest neighbor based on the Euclidean distance. But, the degree of dynamism was very low when compared to the real-time routing scenario. In order to fix this problem, a multilayer spiking neural P system is designed, which will predict the new customer coordinates based on the trained dataset. The trained model computes the results and maps the dynamic customers to appropriate regions. The route optimization in each region is performed by FA. The outline of the proposed system is illustrated in Fig. 4. It contains 2 modules:

- Customer coordinate agent
- Route planning agent

The customer coordinate agent handles the dynamic request from customers. It invokes the SN P system to predict the dynamic coordinate region and this information will be transferred to the route planning agent. The route planning agent performs optimization of routes using FA and then the routes are combined. That is, the function of the route planning agent is to perform optimal routing inside each

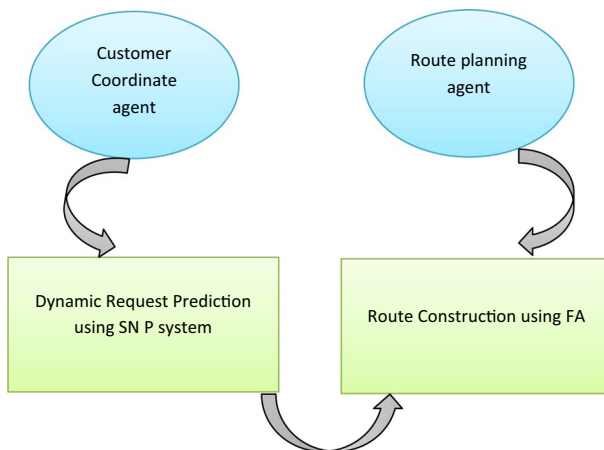


Fig. 4 Agent outline of the proposed system

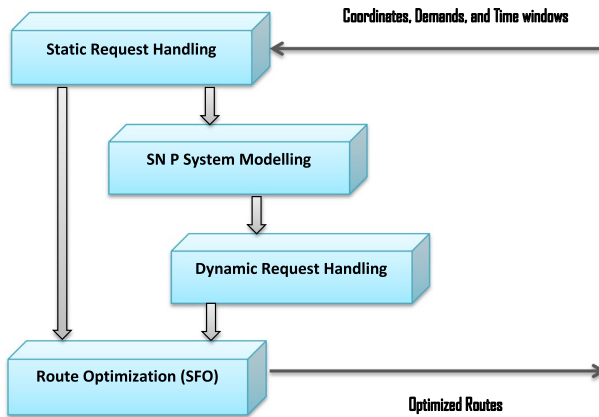


Fig. 5 Block diagram of the proposed system

cluster/region in order to optimize the objectives described in Sect. 4.3. If we use a multi-agent system including vehicle agent, dynamic agent, etc., the performance can be greatly improved. A detailed block diagram of the proposed system is given in Fig. 5. The inputs of the system are customer coordinates, demands of the customers and time windows. After passing this information through all the modules, finally, we get the optimized routes for each of the vehicles as output. A step by step description of the same has been given in the following sections.

6.1 Static clustering of customer coordinates

Here, initially, k means clustering is used to map the known customer coordinates into regions. The travel distance and travel time are also considered. The k value is evaluated for each dataset using the elbow method [59]. After obtaining the result of the elbow method for each dataset, the location of the customer region is obtained by the k -mean method.

6.2 Spiking neural P system modeling

A set of SN P systems with modified rules and training are designed to handle the problem. There are mainly two modules in the proposed SN P system:

- Read module
- Cluster module

Read module: The read module consists of a total of 42 neurons that read the input spike trains to the system. Representation of geo-location coordinates by spike trains is detailed in Sect. 6.3.1. The structure of the SN P system read module is given in Fig. 6. It consists of 42 neurons (40 neurons in general and two neurons,

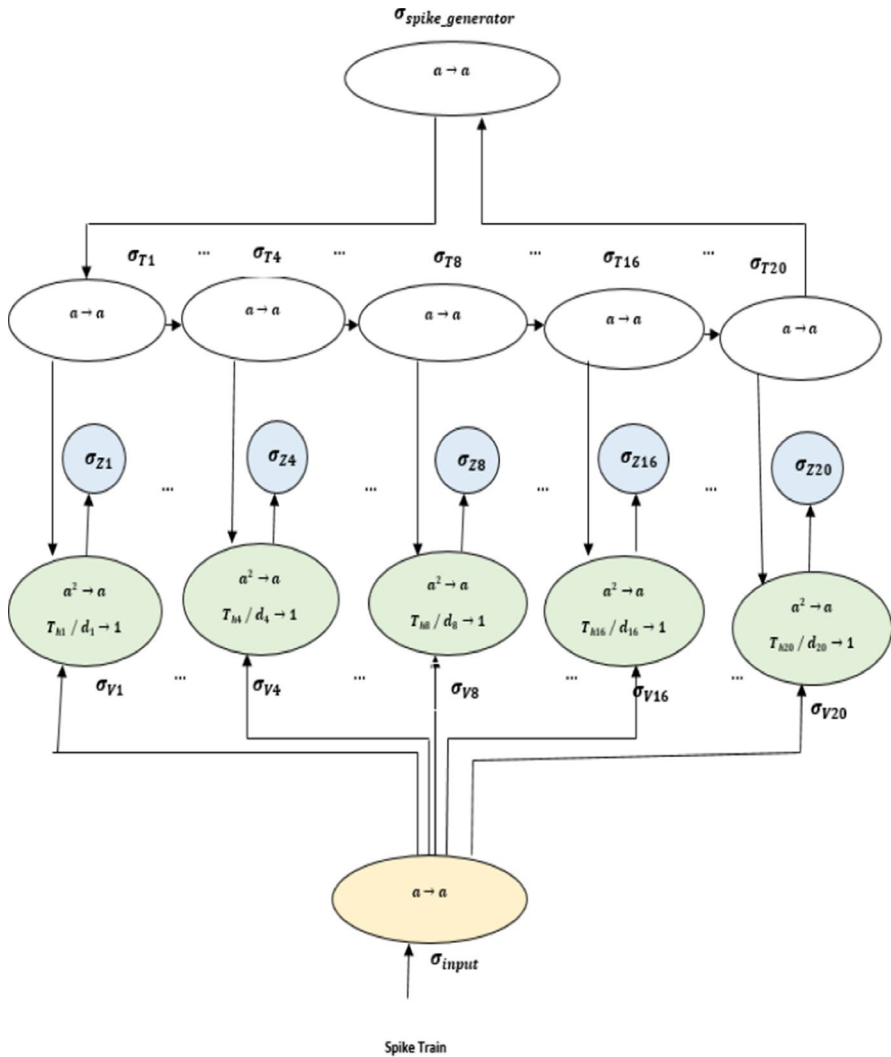


Fig. 6 Spiking neural P system for reading module

one for reading the input spike train and another for spike generation). The process of inputting spike train $s = s_1 s_2 \dots s_{20}$, where, $s_i \in \{0, 1\}$ is as follows:

Initially, all the neurons potential is set to 0 and has no spike, except the $\sigma_{spike-generator}$ which is having one spike at the start and the potential is set to 0.5. The threshold potential for all the neurons in the input module is 1 and is maintained throughout the reading process so that it will be activated whenever the spiking rule is satisfied. So, at t_1 , $\sigma_{spike-generator}$ fires, passing one spike to σ_{r_1} . Simultaneously σ_{input} fires and reads s (s_1) from the input spike train. If $s_1 = 1$, then the σ_{input} spikes, passing one spike to the neurons $\sigma_{v_i}, i \in \{1, 2, \dots, 20\}$.

Also, σ_{r1} fires, passing on a spike to σ_{r2} and σ_{v1} , enabling the spiking rule in σ_{v1} . Then σ_{v1} fires and store a spike in σ_{z1} . In order to clear the previous spikes, we set the potential of neurons to be equal to one until the next activation by σ_{input} . Now, $\sigma_{z1} = 1$, representing $s_1 = 1$.

If $s_1 = 0$ σ_{input} is disabled at step 1. Since it is disabled, σ_{r2} is deactivated by setting the potential equal to 1. So, $\sigma_{z1} = 0$. The spike train s can be read by bits in a step by step and keep it in $\sigma_{zi,i} \in \{1, 2, \dots, 20\}$. When the reading finishes, a spike from σ_{r20} goes back to $\sigma_{spike-generator}$ by passing all the $\sigma_{ri}, i \in \{1, 2, \dots, 20\}$ returns to the initial configuration and is enabled for reading the next spike train. The weight values are set to 1 in the reading stage.

Cluster module: The structure of the cluster module is shown in Fig. 7. The $\sigma_{zi,i} \in \{1, 2, \dots, 20\}$ is used for the design of SN P systems for clustering. The 20 $\sigma_{zi,i} \in \{1, 2, 20\}$ neurons from the output of the reading module together with the k output neurons (here $k=4$), a total of 24 neurons are used in the cluster module. The spiking rule is used to process the information, whereas the spiking activity is controlled by the potential rule. If a neuron contains n spikes and the potential is enabled, it spikes for n times in n steps. A three-layer structure is designed for passing and processing information. The input layer, the intermediate layer, and the output layer. The input layer and the intermediate layer are fully connected. The connections between the intermediate layer and the output layer are specially formulated.

The σ_{z1} and σ_{z10} are chosen as two input neurons, $\sigma_{zi}, i = \{3, 9, 12, 13, 15, 18\}$ as intermediate neurons and the remaining neurons as output neurons. The neurons in the intermediate layer and the outer layer are divided into 2 and 4 groups respectively. There are 12 neurons in the output layer in which each of the 3 neurons is grouped and colored by red, green, blue and yellow respectively. Finally, 4 output neurons are designed to take the information from the 4 groups of output layers. Neuron $\sigma_{zi,i} \in \{1, 2, \dots, 20\}$ will fire for n times if there are n spikes, passing one spike each time. The weight of the synapses in the cluster module is initialized by 1 and updated by the function f_{reLu} (Eq. 22). Thus the 4 output neurons $\sigma_{outi}, i = \{1, 2, 3, 4\}$ have spikes and are being counted. A 4-dimensional vector of a number of spikes (NS) is obtained after finishing the firing process. Also, the potential values of the 4 output neurons are recorded. So, we get a 4 dimensional ordered pair (NS_i, P_i) vector, where $I \in \{1, 2, 3, 4\}$ for each of the input spike train combinations.

For each of the synapses in the cluster module, an activation rule is defined in the form:

(f, d) where,

1. f - activation function as defined in Eq. 22.
2. d =delay

$$= \begin{cases} 0, & \text{if the spikes are emitted immediately} \\ 1, & \text{if the spikes are emitted immediately} \end{cases}$$

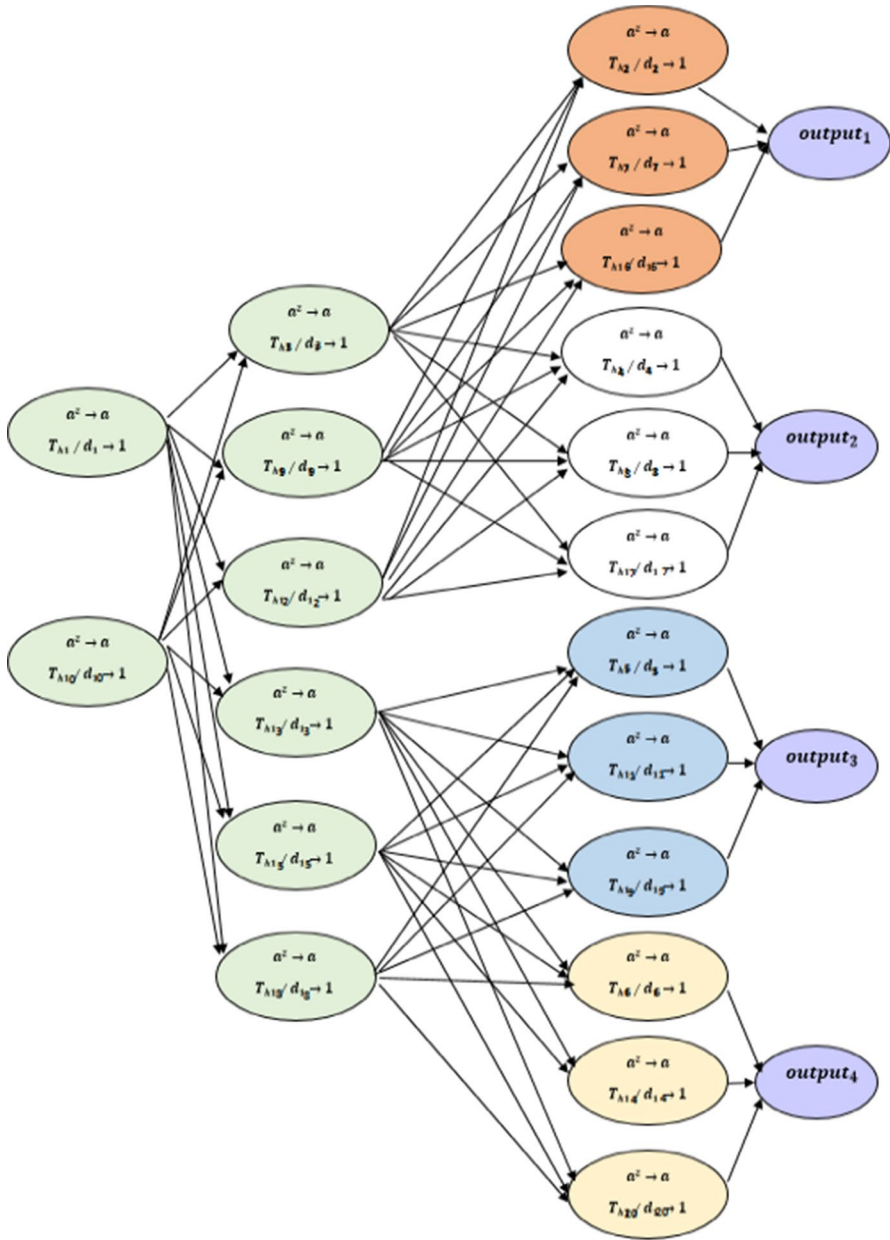


Fig. 7 Spiking neural P system for cluster module

$$f_{\text{reLu}}(x) = \begin{cases} 0, & \text{if } x \leq 0 \\ x, & \text{if } x > 0 \end{cases} \quad (22)$$

6.3 Mapping dynamic coordinates using spiking neural P system

The processes involved in mapping dynamic coordinates are described below.

6.3.1 Encoding customer coordinates by spike trains

In this section, the way to encode customer coordinate values by spike trains is discussed. For each customer coordinate (x, y) , 10 binary digits are used for each of the x and y values. If the coordinates involve fractions, a round to next integer calculation is performed for easier processing. With 10 bits, a maximum of 512 km in both x and y axes can be considered. If one needs to cover more distance, more bits can be added. For example, if a customer is located at $(52, 128)$ then the corresponding spike train will be encoded as:

$$(52, 128) = 0, 000, 110, 1000, 010, 000, 000$$

The most significant 10 digits represent the x coordinate value and the least significant 10 digits represent the y coordinate value of a customer.

6.3.2 The process of mapping

The process of mapping coordinates involves mainly 4 steps. It is depicted in Fig. 8.

Coordinate input: The customer coordinates x and y are given as spike trains to the input neuron σ_{input} , $\sigma_{x_{\text{input}}}$ and $\sigma_{y_{\text{input}}}$ from the environment. Initially, the potential of the input neuron is set to 0.5 and the threshold potential is set to 1. So, it

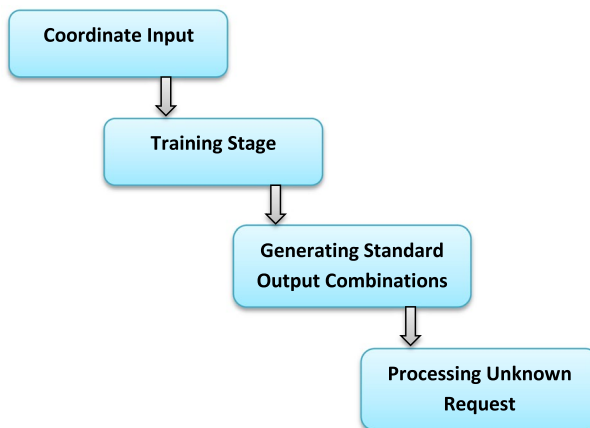


Fig. 8 Block diagram of mapping dynamic requests

is always been enabled and spike. After consuming a fraction of the potential, the remaining potential is emitted to the neighboring neurons with the spike, there it may get added or subtracted. In this way, the input information will be passed from the input layer to the output layer.

Training: When the input module reads the spikes, the $\sigma_{z_i,i} \in \{1, 2, 20\}$ neurons will contain a certain number of spikes. The potential of the neuron is managed by a potential adapter [60]. So, when, $\sigma_{z_i,i} \in \{1, 2, 20\}$ has spikes, it fires if the potential is less than that of the threshold potential. During the firing process, the weight of each of the synapses will be updated by the learning function as defined in Eq. 22. Thus the spikes will be passed through the synapses and thus the potential of all of the participant neurons will be changed. Finally, it reaches the output layer. The ‘ k ’ neurons (for each cluster evaluated by elbow method) in the output layer is associated with a potential and can emit spikes to the environment. At the end of the computation, we obtain a topological structure, which is a set of values in the form of (NS_i, P_i) where $i \in \{1, 2, \dots, k\}$. We run each coordinate input 10 times and compute the average. Thus for k clusters, k trained SN P systems will be obtained. These trained systems will be used to cluster unknown coordinates. Since the size of the training set of each cluster used in the proposed work is within a finite limit w , the bound to the weights on each synapse is not more than w . For SN P systems, the synapse weight should be limited. So the worst-case bound to the weight of the synapse is $O(w^3)$. The difference between known and unknown coordinates inputs are simplified using a variance.

Generating standard output combinations: For an input (x - y coordinates of the customer), the set of spikes trains are read into the k trained SN P systems. With this coordinate input, each of the systems starts computing. Finally, we will get a k dimensional vectors of the form (NS_i, P_i) , where $i \in \{1, 2, \dots, k\}$. We run the coordinates for a maximum of 10 iterations and computed the combination of an average number of spikes generated and the average potential for each of the k output neurons. It is given by:

$$N_{avg} = \sum_{j=1}^{max-iter} \sum_{i=1}^k NS_{ji} / max-iter \tag{23}$$

$$P_{avg} = \sum_{j=1}^{max-iter} \sum_{i=1}^k P_{ji} / max-iter \tag{24}$$

where $max-iter = 10$ is used in the proposed system.

Clustering unknown coordinates: If a new customer with a dynamic request arrives, its coordinate values are read into each of the k trained SN P systems and allow them to start processing. It will produce a k dimensional vector (NS_i, P_i) where $i \in \{1, 2, k\}$ after the computation halts for each of the k output neurons which are the number of spikes and the corresponding potential respectively. Then it calculates the variance between the output vectors of unknown coordinates and known output vectors using the formula,

$$\text{Var}_{\text{NS}} = \sqrt{\sum_{i=1}^k \text{NS}_{\text{unknown}(i)} - N_{\text{avg}(i)}} \quad (25)$$

$$\text{Var}_P = \sqrt{\sum_{i=1}^k P_{\text{unknown}(i)} - P_{\text{avg}(i)}} \quad (26)$$

where $\text{NS}_{\text{unknown}(i)}$, $i \in \{1, 2, \dots, k\}$ is the k output vectors of unknown coordinates and $N_{\text{avg}(i)}$, $i \in \{1, 2, \dots, k\}$ is the average output vector of known trained coordinates. The one with the lowest value of variance is considered as the final result.

6.4 Firefly encoding

The route optimization is done by FA. The main processes are described as follows.

6.4.1 Motivation

Though Most of the existing methods generally perform well in almost all circumstances where they are implemented, there is still room for further modifications of new methods since current methods have some disadvantages in terms of computational complexity or in applications. An optimization approach that is well suited for VRP and its variants is ACS [61]. Authors have tried many combinations with ACS such as ACS with TS [62], PSO [63], and variable neighborhood search (VNS) [64], etc. But major limitations of ACS include premature convergence and getting trapped into local optima.

On the other hand, FA has acquired high popularity; it has been rarely applied in VRP. The process of encoding fireflies in discrete optimization problems is a major issue. But, due to its multimodality nature, more diversified solution space can be obtained that is very useful for routing problems. So, it was felt that appropriate initialization and distance calculation of FA might be more effective in solving routing problems. Keeping this in mind, we adopted a different firefly encoding and distance calculation suitable for the application.

6.4.2 Overview

Although FA was designed for continuous optimization problems, it has been applied to many discrete optimization problems after some modifications. But the quality of solutions depends on the initial pool of feasible solutions. Also in VRP, the duplication of customers may result in the position update strategy of FA. So it involves the maintenance overhead of infeasible solutions.

The initial feasible solution is defined by Clarke and wright algorithm [65]. Also, the fireflies are encoded as a set of customers with index denoting the visiting order.



Fig. 9 Representation of tour 1



Fig. 10 Representation of tour 2

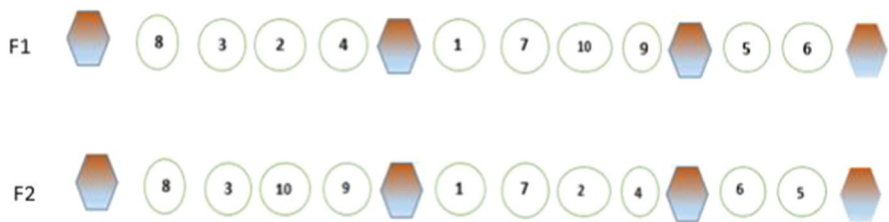


Fig. 11 Firefly F1 and firefly F2

The shaded structures are used to denote the depot. Figures 9 and 10 show different firefly representations of two different tours.

The connection distance C_{ij} between two fireflies is defined by:

$$C_{ij} = \begin{cases} 0, & \text{if node } i \text{ is connected to node } j \text{ in } F1 \text{ then in } F2 \text{ node } i \text{ should be connected to node } j \\ 1, & \text{Otherwise} \end{cases} \tag{27}$$

The total distance between two fireflies is the sum of the connection distances between the two fireflies F1 and F2.

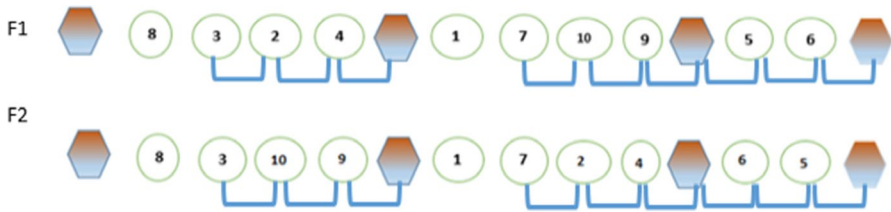


Fig. 12 Distance calculation of two fireflies (individual distance is represented by blue lines)

Figure 11 shows the two different firefly representations F1 and F2. The distance between F1 and F2 is 9 which is the sum of distances of all the connections which are connected to different connections in two fireflies. It is depicted in Fig. 12.

7 Experimental results and discussions

In this section, the performance of the proposed system to solve DGVRPTW will be rigorously assessed. The system is tested on a benchmark dataset of DVRPTW instances and GVRPTW instances separately and the results are analyzed. Various parameter settings are given in Table 1. All the experiments are simulated using MATLAB with Intel xenon 2.93 GHz processor, 12 GB RAM and Windows 10 OS.

7.1 Case study 1: DVRPTW dataset

The proposed system is tested by an average of 40 runs on each of the benchmark dataset, which are open and available at <http://neo.lcc.uma.es/vrp/>. A comparison of the solution quality in terms of the best and average values of the proposed system is done with state-of-the-art schemes: ACO, k-ACO and E-ACO [20], VNS [66], GA-DVRP [67], and is tabulated in Table 2. The best results are in boldface. The proposed system achieves 17 out of 20 best values and 15 out of 20 average values compared with other best-known ACO variants. The K-ACO attains the worst performance compared to the other algorithms in terms of best and average solutions.

The solution improvement percentage of the proposed system is compared with other schemes considering k-ACO as the benchmark scheme. It is noted that the proposed system achieves the highest solution improvement. It is given in Fig. 13.

Table 1 Parameter settings

| Parameter | Value |
|--------------------------|---------|
| α | 0.2 |
| β | 1 |
| γ (Initial value) | 1 |
| Maxiter | 100–300 |
| Population size | 30 |

Table 2 Comparison of the best and the average values

| Problem | ACO | | K-ACO | | E-ACO | | VNS | | GA-DVRP | | Proposed scheme | |
|---------|---------|----------------|---------|---------|---------|---------------|---------|---------|----------------|----------------|-----------------|-----------------|
| | Best | Average | Best | Average | Best | Average | Best | Average | Best | Average | Best | Average |
| i71 | 311.18 | 358.69 | 286.42 | 315 | 259.71 | 297.08 | 304.32 | 325.18 | 288.3 | 319.49 | 210.3 | 322.8 |
| C50 | 631.3 | 681.86 | 611.27 | 652.45 | 607.21 | 647.21 | 599.53 | 653.84 | 566.01 | 597.34 | 679.7 | 585.1 |
| c75 | 1009 | 1042 | 982.30 | 1136.90 | 924.71 | 1045.44 | 981.64 | 1040.00 | 944.46 | 990.78 | 811.5 | 987.4 |
| c100 | 973.26 | 1066.16 | 1021.00 | 1162 | 973 | 1044.96 | 1022.92 | 1087.18 | 943.89 | 988.15 | 1088 | 909.3 |
| c100b | 944.23 | 1023.60 | 921.41 | 1000.80 | 869.22 | 950.17 | 866.71 | 942.81 | 869.41 | 904.03 | 764.6 | 784.5 |
| c120 | 1416.45 | 1525.15 | 1137.50 | 1297 | 1108.15 | 1197.68 | 1285.21 | 1469.24 | 1288.66 | 1399.40 | 1100.1 | 1164.5 |
| c150 | 1345.73 | 145.50 | 1480.00 | 1730.90 | 1378.63 | 1472.40 | 1334.73 | 1441.37 | 1273.50 | 1359.25 | 1261.3 | 1473.4 |
| c199 | 1771.04 | 1844.82 | 1732.00 | 1971 | 1561.1 | 1836.9 | 1679.65 | 1769.95 | 1646.36 | 1700.54 | 1522.5 | 1681.2 |
| tai75a | 1841 | 1945.2 | 1657.10 | 2083.30 | 1690.91 | 1983.92 | 1806.81 | 1954.25 | 1744.78 | 1823.71 | 1481.9 | 1586.1 |
| tai75b | 1535.43 | 1704.06 | 1541.90 | 1680.70 | 1509.6 | 1647.78 | 2250.5 | 2462.50 | 2181.31 | 2290.95 | 1314.4 | 1489.3 |
| tai75c | 1574.98 | 1653.58 | 1395.10 | 1591.60 | 1329.42 | 1470.60 | 3479.44 | 3680.35 | 3280.79 | 3449.32 | 1308.1 | 1312.8 |
| tai75d | 1472.35 | 1529.00 | 1494 | 1663.10 | 1409 | 1661.73 | 1480.7 | 1560.71 | 1441.35 | 1546.18 | 1312.3 | 1799.2 |
| tai100a | 2375.92 | 2428.4 | 2497.6 | 2714.20 | 2281.7 | 2550.61 | 2169.1 | 2319.72 | 2119.03 | 2212.58 | 2289.2 | 2311.5 |
| tai100b | 2283.97 | 2347.90 | 2365.80 | 2624.70 | 2255.83 | 2500.72 | 2934.86 | 3089.57 | 2885.94 | 3073.58 | 2131.4 | 2202.3 |
| tai100c | 1562 | 1655.91 | 1548 | 1764.50 | 1442.45 | 1743.1 | 1621.03 | 1746.07 | 1433.73 | 1502.56 | 1316.1 | 1598.2 |
| tai100d | 2008.13 | 2061 | 1958.80 | 2228 | 1581 | 1844 | 1490.58 | 1557.81 | 1504.63 | 1589.76 | 1398.2 | 1709.3 |
| tai150a | 3644.78 | 3840.18 | 3800.60 | 4216.40 | 3307.63 | 3684.03 | 2674.29 | 2928.77 | 2593.78 | 2759.96 | 2479.22 | 5197.1 |
| tai150b | 3166.88 | 3327.47 | 3270.2 | 3654.80 | 3128.00 | 3439.38 | 1446.50 | 1541.98 | 1408.48 | 1434.56 | 1305.9 | 3119 |
| tai150c | 2311.48 | 3016 | 2795.00 | 2886.20 | 2583.4 | 2729.15 | 1969.94 | 2100.38 | 1793.64 | 1916.03 | 1579.3 | 2498.1 |
| tai150d | 3058.81 | 3203.75 | 2981.20 | 3245.21 | 2809 | 3186.1 | 2954.54 | 3147.88 | 2911.47 | 3010.34 | 2668.4 | 3076.4 |
| average | 1787 | 1885.5 | 1856.83 | 2064.07 | 1650 | 1846.64 | 1717.65 | 2540.06 | 1655.976 | 1743.426 | 1401.121 | 1790.375 |
| Count | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 3 | 3 | 17 | 15 |

The best results are bolded

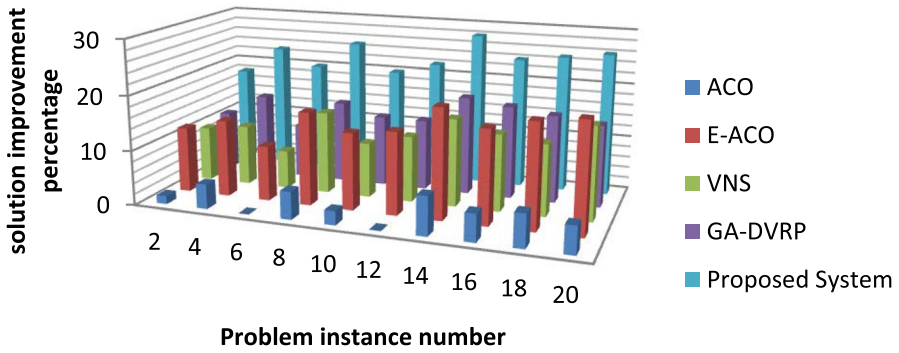


Fig. 13 Comparison of solution improvement percentage of various schemes (K-ACO as the benchmark scheme)

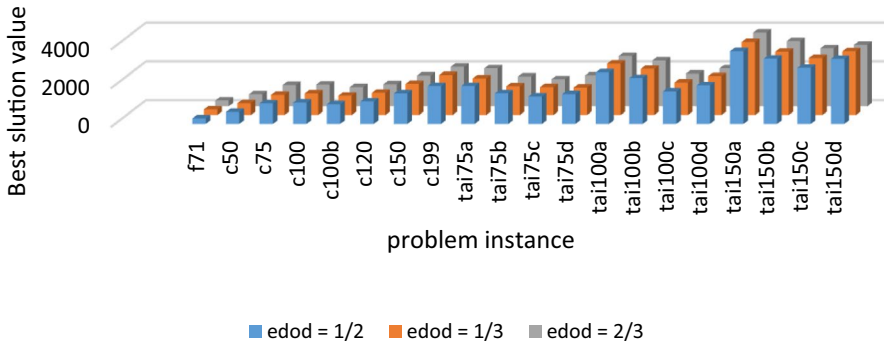


Fig. 14 Comparison of best solutions based on different degrees of dynamism

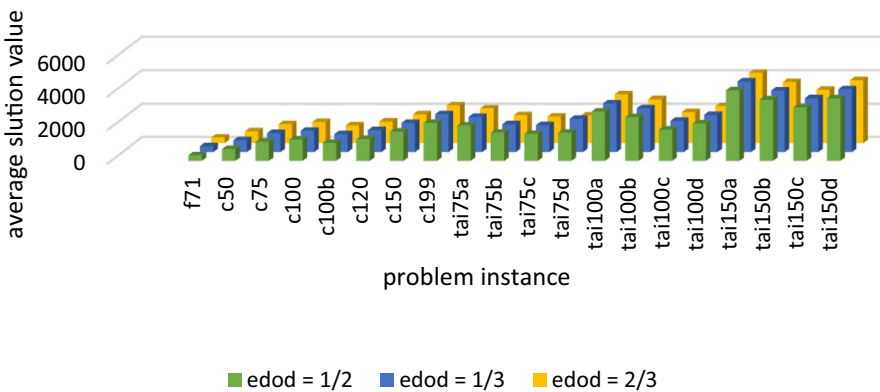


Fig. 15 Comparison of average solutions based on different degrees of dynamism

The effect of dynamism with different dod ($1/3, 1/2$ and $2/3$) under time windows is analyzed. The best and average values under different dod are given in Figs. 14 and 15, respectively. It is noted that the distribution of best and average results is relatively balanced and stable with different edod. So, we concluded that the proposed system is stable and robust.

7.2 Case study 2: GVRPTW dataset

The GVRPTW benchmark dataset is taken from EI bouzekri [68], which is having a range of customers from 10 to 300. The depot is identified as $[0, 0]$ and the customer coordinates are included in the region $[0, 100]$ km. The capacity of the vehicle is set as 25000 kg. The load volumes of customers randomly belong to $[500 \text{ kg}, 2500 \text{ kg}]$ and the service time of customers is fixed at 15 min. The service period should range in $[8 \text{ h}, 18 \text{ h}]$ and the average speed of the vehicle is fixed as 80 km/hr.

To make the static GVRPTW benchmark dataset dynamic, we performed the following:

- Appearance time- for each order, the time of working day, when the request becomes known to the SN P system.
- The number of vehicles is set to 30 for each instance.

We performed a comparison of the influence of dynamism on minimizing CO₂ emissions by vehicles, and the solution quality is compared by minimizing the total emissions at the static and dynamic environment. From the analysis of the result, it is found that the CO₂ emissions and fuel consumption are high in dynamic requests due to the dynamic allocation of vehicles to corresponding customers within the time frame regardless of the location. In static case, the routes have already been known to the depots so no re-rerouting need to be performed once starts serving customers. It is detailed in Fig. 16. The quality of solutions versus CO₂ emissions is analyzed in a static and dynamic environment and is tabulated in Table 3. It is

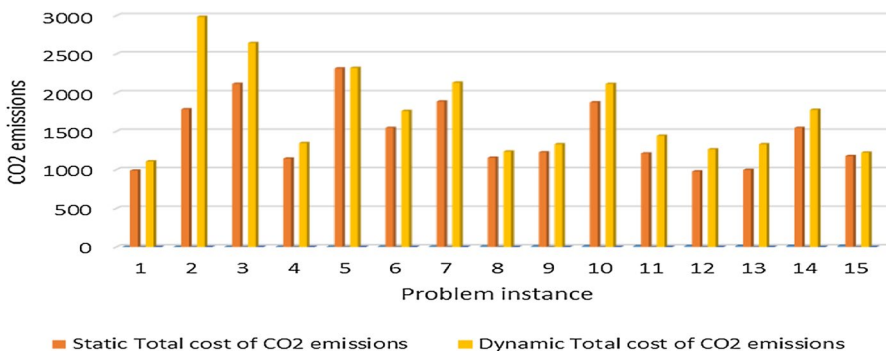


Fig. 16 Analysis of the influence of dynamism in CO₂ emissions

Table 3 Analysis of solution quality versus CO₂ emissions under static and dynamic conditions

| Problem instance | Static | | Dynamic | |
|------------------|---|----------------------|---|----------------------|
| | The total cost of CO ₂ emissions | Solution quality (%) | The total cost of CO ₂ emissions | Solution quality (%) |
| 1 | 987.6 | 94 | 1107 | 92.30 |
| 2 | 1786.4 | 92.10 | 2985 | 89.70 |
| 3 | 2116.8 | 97.40 | 2645 | 98.10 |
| 4 | 1143.7 | 88.50 | 1347.5 | 87.20 |
| 5 | 2312.9 | 99.10 | 2319.7 | 96.10 |
| 6 | 1543.9 | 91.90 | 1764.2 | 90.30 |
| 7 | 1887.4 | 95.30 | 2132.5 | 93.20 |
| 8 | 1154.9 | 83.10 | 1237.3 | 81.20 |
| 9 | 1227.9 | 96.20 | 1332.7 | 95.30 |
| 10 | 1876.1 | 98.70 | 2115.3 | 97.30 |
| 11 | 1211.7 | 91.50 | 1442.6 | 90.20 |
| 12 | 976.9 | 93.90 | 1265.6 | 92.10 |
| 13 | 996.6 | 97.50 | 1332.5 | 95.80 |
| 14 | 1543.8 | 92.80 | 1778.9 | 90.10 |
| 15 | 1176.2 | 92.90 | 1223.2 | 91.30 |

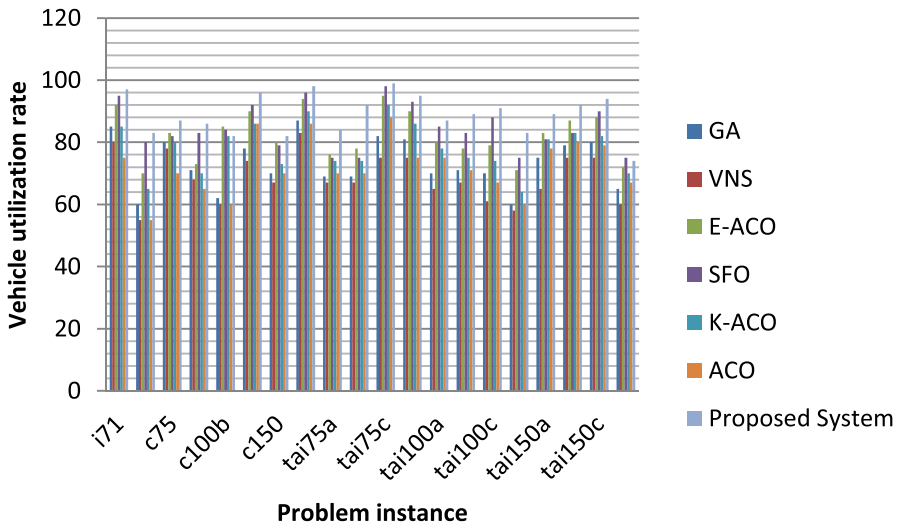


Fig. 17 Comparison of the vehicle utilization rate of different schemes

noted that the solution quality is less for dynamic instances compared to static instances. A comparison of the vehicle utilization rate of various schemes is plotted in Fig. 17.

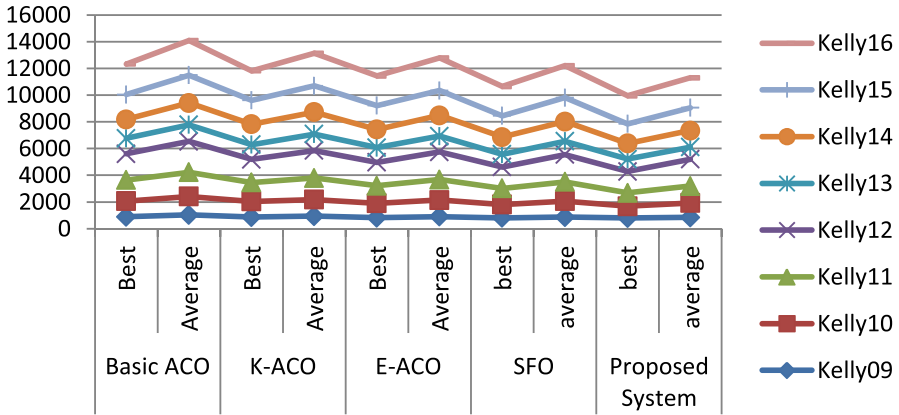


Fig. 18 Comparison of best and average solution values of various schemes on Kelly dataset

7.3 Case study 2: large-scale dataset

Most of the state-of-the-art algorithms use small and medium datasets to test the performance of DVRP. The authors found only one scheme, EACO, which has used the Kelly data, having a scale of 225–480. The SFO [69] algorithm has been simulated on Kelly dataset to compare the performance. Then the proposed system is compared with the ACO variants and SFO on Kelly dataset on best and average solution quality and is given in Fig. 18. From the analysis of the result, it is noted that the proposed scheme is superior in arriving best and average solution values compared to other systems.

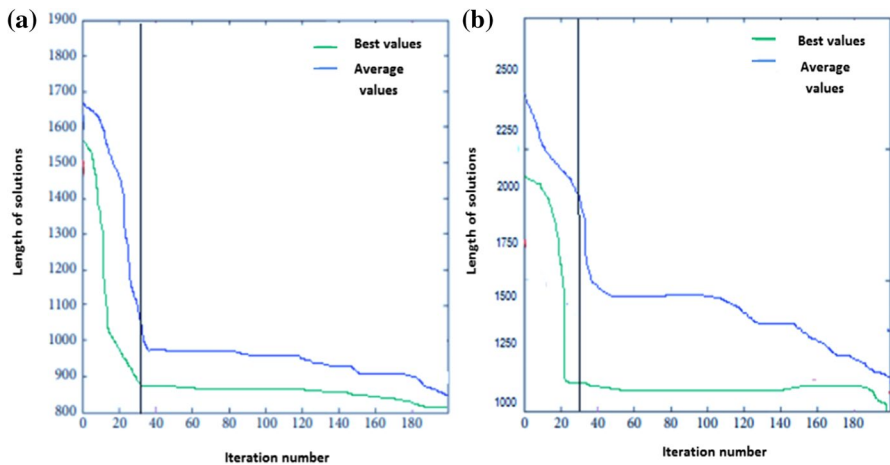


Fig. 19 Nature of convergence of proposed system ((a) Kelly 09 and (b) Kelly 10)

7.4 Nature of convergence of proposed system

The convergence rate of the proposed system has been examined by taking two instances from Kelly Dataset, namely "Kelly09" and "Kelly10," and is given in Fig. 19a and b, respectively. The green line and blue line, respectively, indicate the best and average solution values with the number of iterations. For Kelly 09, the proposed scheme started to converge in 20 iterations while 30 in EACO and SFO [70] schemes. For Kelly 10 instance also the same has noticed. From the analysis of solution values, it has found that the proposed system is superior to other best-known schemes in finding the best and average solutions. So, it is concluded that the proposed system arrives at the optimal solution in a short time.

1. For problems having around 5001000 scale datasets, it is found that the proposed scheme can find optimal solutions in 30 iterations.
2. After analyzing the results, the following conclusions have been gained:
3. The performance of the proposed system is superior to the schemes ACO, KACO, EACO, VNS, and GADVRP in terms of solution values (Table 2).
4. The proposed system is having the best vehicle utilization over other schemes (Fig. 17).
5. The solution improvement percentage of the proposed method is almost 30%, while other schemes have below 20 improvements (Fig. 13).
6. The proposed scheme works well for large-scale datasets as well compared to existing schemes in terms of solution values (Fig. 18).
7. It is noted that the distribution of best and average results is relatively balanced and stable with different *edod*. So, we concluded that the proposed system is durable and robust (Figs. 14, 15).
8. It is scalable since it shows excellent performance in large-scale datasets. Also, the convergence rate is high (20 iterations) compared to existing schemes (Fig. 19).

Table 4 Pairwise T-test statistics

| | | Mean | <i>N</i> | Std. deviation | Std. error mean | Sig |
|--------|-----------------|-----------|----------|----------------|-----------------|--------|
| Pair 1 | ACO | 1761.8915 | 20 | 857.2475 | 191.6864 | 0.003 |
| | Proposed scheme | 1419.395 | 20 | 605.8148 | 135.4643 | |
| Pair 2 | E-ACO | 1650.4835 | 20 | 831.3698 | 185.8999 | 0.000 |
| | Proposed scheme | 1419.395 | 20 | 605.8148 | 135.4643 | |
| Pair 3 | K-ACO | 1984.8967 | 20 | 975.99005 | 212.97849 | 0.000 |
| | Proposed scheme | 1419.395 | 20 | 605.8148 | 135.46 | |
| Pair 4 | GA-VRP | 1655.976 | 20 | 803.84 | 107.025 | 0.0395 |
| | Proposed scheme | 1419.395 | 20 | 605.8148 | 135.46 | 0.0173 |
| Pair 5 | VNS | 1717.6410 | 20 | 830.7174 | 185.75 | |
| | Proposed scheme | 1419.395 | 20 | 605.8148 | 135.46 | 0.0173 |

Table 5 Holm–Bonferroni ranking (proposed scheme as the reference with rank 5.15)

| | RANK | ZJ | PJ | D/J | output |
|---------|------|----------|----------|--------|----------|
| E-ACO | 4.85 | −2.08333 | 0.01401 | 0.0185 | Rejected |
| K-ACO | 4 | −3.43137 | 0.007143 | 0.025 | Rejected |
| GA-DVRP | 3 | −5.26961 | <0.00001 | 0.05 | Rejected |
| VNS | 2.8 | −5.4321 | <0.00001 | 0.024 | Rejected |
| ACO | 2 | −5.1098 | <0.00001 | 0.0328 | Rejected |

7.5 Statistical analysis

A statistical analysis has been performed over the algorithms considered for DVRPTW datasets. A pairwise t-test has been done to check the significance of the proposed scheme. The null hypothesis for the paired t-test is assumed as the true mean difference between paired samples is zero. On the other hand, the alternative hypothesis is that the true mean difference between paired samples is not zero. The level of significance is 0.05. Since all pairs have significantly less than 0.05, the null hypothesis is rejected. The results of the pairwise t-test are given in Table 4. The results show that there are significant differences between these pairs of algorithms.

After analyzing the results so far, the algorithms are ranked on the basis of the average solution quality obtained over 20 test instances. A score is assigned for each of the problem instances as per the Holm-Bonferroni [70] ranking scheme. The p values are compared with the calculated values and analyzed the system performance. The results are given in Table 5. The significance is given as ‘Accepted’ if the null hypothesis (no difference in performance between the considered algorithms) is true otherwise ‘Rejected’ if the null hypothesis is false.

The results summarized above appear competitive with respect to state-of-the-art algorithms.

8 Conclusions and future scope

As spiking neural P systems are computing models, having maximum exploration and exploitation capabilities, they are adequate for arriving global optimal solutions in a feasible time. In this paper, we proposed a multilayer SN P system with potentials and activation functions, which does region mapping of customers in DGVRPTW, followed by a route optimization by FA. So, the major concern is to focus on improving dynamic measures which is well tackled by the proposed scheme. Apparently, other measures such as convergence rate and vehicle utilization have also been improved. It is tested on benchmark instances and proved the efficiency and reliability of the system.

The advantages of proposed system are as follows:

1. The performance of the proposed system is competitive to the other schemes in terms of best and average solution quality.
2. The solution improvement percentage of the proposed system is high compared to state-of-the-art schemes.
3. The quality of solutions is high under static conditions, and the amount of CO₂ emissions is found to be less.
4. The distribution of best and average results is the same under different *dod*, and thereby, the system is stable and robust.
5. A balance between solution quality and time impediments has been found on large-scale dataset, which highlights the scalability of the system

Even though the system works better than state-of-the-art schemes, there is still room for further improvements in terms of number of vehicles and distance metric. As future research directions, we may focus on improving the solution quality further along with other optimization constraints as well.

References

1. Popovici E, Bucci A, Wiegand RP, de Jong ED (2012) Handbook of natural computing. Springer, Berlin
2. Agatonovic-Kustrin S, Beresford R (2000) Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. *J Pharma Biomed Anal* 22:717–727. [https://doi.org/10.1016/S0731-7085\(99\)00272-1](https://doi.org/10.1016/S0731-7085(99)00272-1)
3. Zhang Y, Agarwal P, Bhatnagar V, Balochian S, Yan J (2013) Swarm intelligence and its applications. *Sci World J* 2013:3. <https://doi.org/10.1155/2013/528069>
4. Bartz-Beielstein T, Branke J, Mehnen J, Mersmann O (2014) Evolutionary algorithms. *Wiley Interdiscip Rev Data Min Knowl Discov*. <https://doi.org/10.1002/widm.1124>
5. Noson Y (2007) An introduction to quantum computing. Springer, Dordrecht
6. Păun G (2010) A quick introduction to membrane computing. *J Logic Algebra Program* 79(6):291–294. <https://doi.org/10.1016/j.jlap.2010.04.002>
7. Păun G, Rozenberg G (2002) A guide to membrane computing. *Theor Comput Sci* 287:73–100. [https://doi.org/10.1016/S0304-3975\(02\)00136-6](https://doi.org/10.1016/S0304-3975(02)00136-6)
8. Song B, Pan L, Pérez-Jiménez MJ (2016) Cell-like p systems with channel states and symport/antiport rules. *IEEE Trans Nanobiosci* 15(6):555–566
9. doi: <https://doi.org/10.1109/TNB.2016.2594192>
10. Pan L, Pérez-Jiménez MJ (2010) Computational complexity of tissue-like P systems. *J Complex* 26(3):296–315. <https://doi.org/10.1016/j.jco.2010.03.001>
11. Wang X et al (2016) On the computational power of spiking neural P systems with self-organization. *Sci Rep* 6:27624. <https://doi.org/10.1038/srep27624>
12. Zhao Y, Liu X, Wang W (2016) Spiking neural P systems with neuron division and dissolution. *PLoS ONE* 11:e0162882. <https://doi.org/10.1371/journal.pone.0162882>
13. Pan L, Păun G, Pérez-Jiménez M (2011) Spiking neural P systems with neuron division and budding. *Sci China Inf Sci* 54:1596–1607. <https://doi.org/10.1007/s11432-011-4303-y>
14. Song Jun Wang X, Ning Zhang Sun HPG, Fan Yang T (2018) Spiking neural P systems with multiple channels and anti-spikes. *Biosyst Vol* 169–170:13–19. <https://doi.org/10.1016/j.biosyst.2018.05.004>
15. Chen H, Freund R, Ionescu M, Paun G, Pérez-Jiménez M (2007) On string languages generated by spiking neural P systems. *Fundam Inform* 75:141–162
16. Zhang G, Rong H, Neri F, Pérez-Jiménez M (2014) An optimization spiking neural P system for approximately solving combinatorial optimization problems. *Int J Neural Syst* 24:1440006. <https://doi.org/10.1142/S0129065714400061>

17. Díaz-Pernil D, Gutiérrez-Naranjo M (2017) Semantics of deductive databases with spiking neural P systems. *Neurocomputing*. <https://doi.org/10.1016/j.neucom.2017.07.007>
18. Diaz C, Frias T, Sanchez G, Perez-Meana H, Toscano K, Duchon G (2017) A novel parallel multiplier using spiking neural P systems with dendritic delays. *Neurocomputing*. <https://doi.org/10.1016/j.neucom.2017.02.009>
19. Tingfang Wu, Wang Y, Jiang S, Yansen Su, Shi X (2018) Spiking neural P systems with rules on synapses and anti-spikes. *Theor Comput Sci* 724:13–27. <https://doi.org/10.1016/j.tcs.2017.12.015>
20. Clarke G, Wright JW (1964) Scheduling of vehicles from a central depot to a number of delivery points. *Oper Res* 12(4):568–581
21. Jean-François C, Guy D, Jacques D, François S (2002) VRP with time windows. Society for Industrial and Applied Mathematics, Philadelphia
22. Larsen A (2000) The dynamic vehicle routing problem. Technical University of Denmark, Kgs. Lyngby, Denmark
23. Erdoğan S, Miller-Hooks E (2012) A green vehicle routing problem. *Transp Res Part E Logist Transp Rev* 48(1):100–114. <https://doi.org/10.1016/j.tre.2011.08.001>
24. Arora S, Singh S (2013) The firefly optimization algorithm: convergence analysis and parameter selection. *Int J Comput Appl* 69(3):10
25. Gandomi A, Yang X, Alavi A (2011) Mixed variable structural optimization using a firefly algorithm. *Comput Struct* 89(23):2325–2336
26. Gao M, He X, Luo D, Jiang J, Teng Q (2013) Object tracking using firefly algorithm. *IET Comput Vision* 7(4):227–237
27. Crainic TG, Gendreau M, Potvin J-Y (2009) Intelligent freight-transportation systems: assessment and the contribution of operations research. *Transp Res Part C Emerg Technol* 17(6):541–557. <https://doi.org/10.1016/j.trc.2008.07.002>
28. Attanasio A, Bregman J, Ghiani G, Manni E (2007) Real-time fleet management at Ecourier Ltd. In: Zempeki V, Tarantilis CD, Giaglis GM, Minis I (eds) *Dynamic fleet management of operations research/computer science interfaces*. Springer, Boston
29. Caramia M, Italiano G, Oriolo G, Pacifici A, Perugia A (2002) Routing a fleet of vehicles for dynamic combined pick-up and deliveries services. In: *Proceedings of the symposium on operation research 2001*, Duisburg, Germany
30. Beaudry A, Laporte G, Melo T, Nickel S (2010) Dynamic transportation of patients in hospitals. *OR Spectr* 32:77–107. <https://doi.org/10.1007/s00291-008-0135-6>
31. Kergosien Y, Lenté C, Piton D, Billaut J-C (2011) A tabu search heuristic for the dynamic transportation of patients between care units. *Eur J Oper Res*. <https://doi.org/10.1016/j.ejor.2011.04.033> (In Press)
32. Romero M, Sheremetov L, Soriano A (2007) A genetic algorithm for the pickup and delivery problem: an application to the helicopter offshore transportation. *Theoretical advances and applications of fuzzy logic and soft computing, advances in soft computing*. Springer, Berlin/Heidelberg, pp 435–444
33. Taniguchi E, Thompson R (2002) Modeling city logistics. *Transp Res Rec J Transpo Res Board* 1790(1):45–51
34. Barcelo J, Grzybowska H, Pardo S (2007) Vehicle routing and scheduling models, simulation and city logistics. In: Zempeki V, Tarantilis CD, Giaglis GM, Minis I (eds) *Dynamic fleet management, operations research/computer science interfaces*, vol 38. Springer, Boston, pp 163–195
35. Zempeki V, Minis I, Mamassis K, Giaglis GM (2007) Dynamic management of a delayed delivery vehicle in a city logistics environment. In: Zempeki V, Tarantilis CD, Giaglis GM, Minis I (eds) *Dynamic fleet management, operations research/computer science interfaces series*. Springer, Boston, US, pp 197–217
36. Gendreau M, Guertin F, Potvin J-Y, Séguin R (2006) Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-up and deliveries. *Transp Res Part C Emerg Technol* 14(3):157–174. <https://doi.org/10.1016/j.trc.2006.03.002>
37. Ghiani G, Manni E, Quaranta A, Triki C (2009) Anticipatory algorithms for same-day courier dispatching. *Transp Res Part E Logist Transp Rev* 45(1):96–106. <https://doi.org/10.1016/j.tre.2008.08.003>
38. Bieding T, Görtz S, Klose A (2009) On line routing per mobile phone a case on subsequent deliveries of newspapers. In: Nunen JA, Speranza MG, Bertazzi L (eds) *Innovations in distribution*

- logistics, lecture notes in economics and mathematical systems, vol 619. Springer, Berlin, Heidelberg, pp 29–51
39. Campbell A, Savelsbergh M (2005) Decision support for consumer direct grocery initiatives. *Transp Sci* 39(3):313–327
 40. Ferrucci F, Bock S, Gendreau M (2011) Real-time distribution of perishable goods using past request information to anticipate future requests. *Oper Res* 34:84
 41. Azi N, Gendreau M, Potvin JY (2011) A dynamic vehicle routing problem with multiple delivery routes. *Ann Oper Res* 13:103–112 (**In-press**)
 42. Balev S, Guinand F, Lesauvage G, Olivier D (2009) Dynamical handling of straddle carriers activities on a container terminal in uncertain environment—a swarm intelligence approach. In: Proceedings of the 2009 international conference on complex systems and applications (ICCSA 2009), University of Le Havre, Le Havre, France
 43. Berbeglia G, Cordeau J-F, Laporte G (2010) Dynamic pickup and delivery problems. *Eur J Oper Res* 202(1):8–15. <https://doi.org/10.1016/j.ejor.2009.04.024>
 44. Fiegl C, Pontow C (2009) Online scheduling of pick-up and delivery tasks in hospitals. *J Biomed Inform* 42(4):624–632. <https://doi.org/10.1016/j.jbi.2009.02.003>
 45. Larsen A, Madsen OBG, Solomon MM (2004) The a priori dynamic traveling salesman problem with time windows. *Transp Sci* 38(4):459–472. <https://doi.org/10.1287/trsc.1030.0070>
 46. Tagmouti M, Gendreau M, Potvin J-Y (2011) A dynamic capacitated arc routing problem with time-dependent service costs. *Transp Res Part C Emerg Technol* 19(1):20–28. <https://doi.org/10.1016/j.trc.2010.02.003>
 47. Lin C, Choy KL, Ho GTS, Chung SH, Lam HY (2014) Survey of green vehicle routing problem: past and future trends. *Expert Syst Appl* 41(4):1118–1138. <https://doi.org/10.1016/j.eswa.2013.07.107>
 48. Erdoğyan S, Miller-Hooks E (2012) A green vehicle routing problem. *Transp Res Part E Logist Transp Rev* 48(1):100–114
 49. Kara I, Kara BY, Yetis MK (2007) Energy minimizing vehicle routing problem. In: International conference on combinatorial optimization and applications (pp. 62–71). Springer, Berlin, Heidelberg
 50. Apaydin O, Gonullu MT (2008) Emission control with route optimization in solid waste collection process: a case study. *Sadhana* 33(2):71–82
 51. Hsueh C-F (2017) A vehicle routing problem with consideration of green transportation. *J Manag Sustain* 7:89. <https://doi.org/10.5539/jms.v7n4p89>
 52. Afshar-Bakeshloo M, Mehrabi A, Safari H et al (2016) A green vehicle routing problem with customer satisfaction criteria. *J Ind Eng Int* 12:529–544. <https://doi.org/10.1007/s40092-016-0163-9>
 53. Yiyong X, Abdullah K (2015) Green vehicle routing problem with time varying traffic congestion. *Transp Res Part E Logist Transp Rev*. <https://doi.org/10.1287/ics.2015.0010>
 54. Abdoli B, MirHassani SA, Hooshmand F (2019) On different formulations of green vehicle routing problem. *J Inf Optim Sci* 40(4):883–903. <https://doi.org/10.1080/02522667.2018.1460137>
 55. Normasari NME, Yu VF, Bachtiyar C, Sukoyo A (2019) A simulated annealing heuristic for the capacitated green vehicle routing problem. *Math Probl Eng* 2019:18. <https://doi.org/10.1155/2019/2358258>
 56. Schneider M, Stenger A, Goeke D (2014) The electric vehicle-routing problem with time windows and recharging stations. *Transp Sci* 48:500–520. <https://doi.org/10.1287/trsc.2013.0490>
 57. Johari N, Zain A, Mustaffa N, Udin A (2013) Firefly algorithm for optimization problem. *Appl Mech Mater*. <https://doi.org/10.4028/www.scientific.net/AMM.421.512>
 58. Turkensteen M (2017) The accuracy of carbon emission and fuel consumption computations in green vehicle routing. *Eur J Oper Res*. <https://doi.org/10.1016/j.ejor.2017.04.005>
 59. Qi F, Liu M (2018) Optimization Spiking Neural P System for Solving TSP. In: Gu X, Liu G, Li B (eds) Machine learning and intelligent communications. MLICOM 2017. Lecture notes of the institute for computer sciences, social informatics and telecommunications engineering, vol 227. Springer, Cham
 60. Syakur MA, Khotimah BK, Rochman EMS, Satoto BD (2018) Integration K-means clustering method and elbow method for identification of the best customer profile cluster. In: IOP conference series: materials science and engineering, 336(1): 012017 IOP Publishing
 61. Ramachandran Pillai R, Arock M (2019) An adaptive spiking neural P system for solving vehicle routing problems. *Arab J Sci Eng* 45:1–17

62. Gajpal Y, Abad P (2009) An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup. *Comput Oper Res* 36(12):3215–3223
63. Zhang H, Zhang Q, Ma L, Zhang Z, Liu Y (2019) A hybrid ant colony optimization algorithm for a multi-objective vehicle routing problem with flexible time windows. *Inf Sci* 490:166–190. <https://doi.org/10.1016/j.ins.2019.03.070>
64. Kao Y, Chen MH, Huang YT (2012) A hybrid algorithm based on ACO and PSO for capacitated vehicle routing problems. *Math Prob Eng* 2012:53
65. Kalayci CB, Kaya C (2016) An ant colony system empowered variable neighborhood search algorithm for the vehicle routing problem with simultaneous pickup and delivery. *Expert Syst Appl* 66:163–175. <https://doi.org/10.1016/j.eswa.2016.09.017>
66. Xu H, Pu P, Duan F (2018) Dynamic vehicle routing problems with enhanced ant colony optimization. *Discret Dyn Nat Soc*. <https://doi.org/10.1155/2018/1295485>
67. Stenger A, Vigo D, Enz S, Schwind M (2013) An adaptive variable neighborhood search algorithm for a vehicle routing problem arising in small package shipping. *Transp Sci* 47(1):64–80
68. Elhassania M, Jaouad B, Ahmed EA (2014) Solving the dynamic vehicle routing problem using genetic algorithms. In: *International conference on logistics operations management*, Rabat, pp. 62–69
69. El Bouzekri El Idrissi E, Elhilali Alaoui A (2014) Evolutionary algorithm for the bi-objective green vehicle routing problem. *Int J Sci Eng Res* 5(9):70
70. RamachandranPillai R, Arock M (2020) Spiking neural firefly optimization scheme for the capacitated dynamic vehicle routing problem with time windows. *Neural Comput and Applic*. <https://doi.org/10.1007/s00521-020-04983-8>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Resmi Ramachandranpillai¹  · Michael Arock¹

✉ Resmi Ramachandranpillai
resmiramachandranpillai@gmail.com

Michael Arock
michaelarock@gmail.com

¹ Department of Computer Applications, NIT, Tiruchirappalli, India