Shortest Path Search in Dynamic Reliability Space: Hierarchical Coloured Petri Nets Model and Application to a Pipeline Network

Hela Kadri
LIP2 Laboratory,
University of Tunis El Manar
Tunis, Tunisia
Email: Hela.Kadri@isi.rnu.tn

Belhassen Zouari
LIP2 Laboratory,
University of Tunis El Manar
Tunis, Tunisia
Email: Belhassen.Zouari@fst.rnu.tn

Abstract—This work proposes a hierarchical Coloured Petri Nets (CPN) model for finding the most reliable path in an oil pipeline network. A path is achieved with the opening of valves along it linking the pipes and with the closing of the adjacent ones to isolate it from the rest of the pipeline network. This work has two goals: firstly to calculate the dynamic reliability of the valves engaged to open (openers) and to close (closers) in the path’s search according to their behavior, and secondly to find an optimal path in the reliability space using Dijkstra's algorithm. Real operational constraints and goals in the seaport are modeled with data extracted from OREDA Database. The results underline how the proposed solution leads to a safer use of the network by avoiding the riskiest valves.

Keywords—Dynamic reliability; Discrete event system; Hierarchical Coloured Petri Nets; Pipeline Network; Dijkstra's algorithm; Simulation.

I. INTRODUCTION

Oil pipeline networks can be installed above the ground, under the ground, or underwater and are characterized by complex facilities with multiple components (tubes, pumps, valves, etc.) submitted to heavy working conditions that lead usually to a decrease in their reliability. The failure prevention in these networks has become an obsession because although every pipeline company is working to properly manage her facilities and to achieve incident-free operations, accidents do happen with serious economic, financial and environmental impacts. Dynamic reliability, basing on relevant information about the state of the pipeline along the time (monitoring) allows us an estimate of its behavior in the future (prognosis). In this context, we are interested in the search of the most reliable path using valves’ dynamic reliability and Dijkstra’s algorithm to minimize the risk of failing to transfer oil while improves also the system performance by avoiding delays and losses due to the eventual failures.

More in detail, firstly, we adopted an embedded formalism to model two important elements of dynamic reliability: component disturbance due to time on the one hand, and operating conditions on the other hand and we applied it to the riskiest component to failure in the oil pipeline network: the valve [12];

Secondly, in these pipelines, the oil transfer is carried out by selecting an alignment (i.e. path) linking the two elements of interest and enabling oil flow by opening the valves in the alignment and closing all adjacent valves in order to isolate it to avoid oil mixture. Let us call openers the valves that must be opened, and closers the valves that must be closed. Using Dijkstra’s algorithm, we determine the pipes composing the most reliable path knowing the dynamic reliabilities of their openers and closers.

To show the impact of the proposed solution, a model of a path search algorithm in a pipeline network has been developed in a hierarchical Coloured Petri Net (CPN) framework aiming to achieve the following goals:

- developing the embedded formalism to model the dynamic reliability;
- representing pipeline network and avoiding the problem of state space explosion for real case;
- defining a clear mechanism able to determine the opener and the closers of each pipe engaged in the search algorithm.

In the following sections we present the developed work in its different stages and aspects.

II. DYNAMIC RELIABILITY

Dynamic reliability concepts have gained increasing popularity as an efficient indicator to assess performance degradation during the system’s life. Time dependent failure rates, as well as impact of operating constraints, cannot be ignored to characterize service efficiency. In industry, nowadays, it is well accepted that constant failure rates, respecting Markovian hypothesis, are no more sufficient to characterize resource confidence [7]. It is also considered that some influence factors determine the evolution of failure distribution laws [6], denoting the natural impact of the environment on the duration of life.

In this paper, we reconsider this concept by applying to valves (Eq.2) in order to calculate the path reliability (Eq.1) used to evaluate the candidate paths of the Dijkstra’s algorithm.
Definition 1: Path reliability \( R_p(t, Z) \) is calculated as the product of dynamic reliability of \( n \) involved valves (i.e. its openers and closers).

\[
R_p(t, Z) = \prod_{i=1}^{n} R(t, Z).
\]  

(1)

Based on this definition and due to the dynamic reliability is bounded by 1, we can remark that the path reliability increases when a minimal number of valves is considered. So more we reduce the number of valves more we have a safer operation.

Definition 2: Valve’s dynamic reliability \( R(t, Z) \) is determined from the conventional expression depending of the dynamic failure rate \( \lambda(t, Z) \) [2].

\[
R(t, Z) = e^{-\int_0^t \lambda(\tau, Z)d\tau}
\]

(2)

where

- \( t_0 \) and \( t \) are respectively the initial instant of functioning and the date of the failure occurrence,
- \( Z \) represents the set of influence factors. The choice of the influence factors depend on the application.

Definition 3: The dynamic failure rate \( \lambda(t, Z) \) depend on time \( t \) and influence factors \( Z \).

\[
\lambda(t, Z) = \lambda_0(t) \times g(Z)
\]

(3)

Definition 4: The failure rate base \( \lambda_0(t) \) which illustrated in Fig. 1, is modeled by the Weibull distribution with two parameters.

\[
\lambda_0(t) = \frac{\beta}{\eta} \times \left( \frac{t}{\eta} \right)^{\beta-1}
\]

(4)

where

- \( \beta \): the shape parameter, unitless;
- \( \eta \): the scale parameter in units of time.

![Figure 1. Bath-Tub Shale of the Failure Rate.](image)

Definition 5: The influence function \( g(Z) \) represents system’s external and internal characteristics.

\[
g(Z) = e^{B \times Z} = e^{\sum_{k=1}^{m} b_k \times z_k}
\]

(5)

where

- \( m \): the number of influence factors taken in the model;
- \( B = (b_1, \cdots, b_m) \): coefficients’ vector of the Cox model [4];
- \( Z = (z_1, \cdots, z_m) \): influence factors’ vector.

Estimation of the dynamic reliability parameters:
The coefficients of the dynamic reliability function were determined through the calculi detailed in [2]. The input data needed to identify coefficients were obtained from a database of measurements generated from OREDA database reliability [12] which is a data collection from various industries. In fact, OREDA provides a representation of the failure rate with a normal distribution characterized by its mean and standard deviation. So, firstly, we randomly generated a sufficient number of failure rate values with MATLAB software environment [1]. We use these values of failure rate to find the correspondent time and influence factors using the Naval Surface Warfare Center (NSWC) approach [10].

III. PATH SEARCH ALGORITHM IN A PIPELINE NETWORK

The case study is a pipeline network representative of an oil-exporting seaport. Previous work related to the case study includes some approaches on alignment selection and scheduling based on a directed graph; [8] can be consulted for alignment selection minimizing interferences with envisaged operations in the network, and [9] for scheduling transfer operations on a flow network within a given maintenance framework.

A. System Description

The pipeline network studied is composed by a set of pipes linking a set of tanks storing oil to a set of loading arms placed at the docks of the seaport. Loading arms are connected to tankers to transport oil to different destinations. Alignments are established using valves: some valves are opened, the openers, along a path linking the elements, whereas some other valves are closed, closers, around the path, isolating the alignment from the rest of the pipe network. Usually the tankers request only one type of oil, therefore an alignment is made between a tanker requesting some type of oil and a tank containing that specified oil. Some alignments also need a pump to overcome the negative pressure difference between the delivering element and the receiving element. Valves and pumps are the only elements that can be controlled, thus they are the basis of our model. Valves are, most of the time, in one of two different states: opened or closed, whereas pumps can be on or off. The transitions between these states can be considered instantaneous when compared with the time spent in any of the states. Fig. 2 depicts a simplified example of oil seaport example and Fig. 3 shows its model as an undirected graph in which arcs represent the valves. The nodes represent pipeline segments (pipe, tank, loading arm) containing each a pump allowing to transport the oil in the two directions. To satisfy requests, this work aims to find every time the most reliable path based on dynamic reliability of the openers and the closers. For each valve, the studied influence factors \( z_i \)
which are components of \(Z = (C_s, T, S)\) influence vector are:

- \(C_s\): Commutation stress, the total number of changing state from open to close or the contrary.
- \(T\): The operating time in opened and in closed states.
- \(S\): The last valve state (opened or closed).

From (5), considering these three influence factors, we can write

\[
g(Z) = e^{(b_1 \ast C_s + b_2 \ast T + b_3 \ast S)}. \tag{6}
\]

The constant \(b_3\) is also called the gamma factor i.e. probability of failure (to close/open)[11].

**B. Path search algorithm**

Different algorithms have been proposed to find the optimal routes. Dijkstra’s algorithm is probably the best known; it is a graph search algorithm that solves the single-source shortest path problem for a graph with nonnegative edge costs, producing a shortest path tree [5].

**Application of the Dijkstra’s algorithm:** Let the path reliability (PR) to a pipeline segment (PS) \(Y\) be the PR from the initial PS to it. Dijkstra’s algorithm will determine some initial PR obtained from the dynamic reliabilities of their correspondent openers and closers and will try to improve them step-by-step as follows:

1) Set to 1 our initial PS.
2) Mark all PS as unvisited. Set initial PS as current.
3) For current PS, consider all its unvisited neighbors, calculate their valves’ dynamic reliabilities and update their PR (from the initial PS).
4) When we are done considering all neighbors of the current PS, mark it as visited. A visited PS will not be checked ever again; its PR recorded now is final and minimal.
5) Set the unvisited PS with the smallest PR (from the initial PS) as the next “current node” and continue from step 3).

With these steps, the shortest reliable path from the starting PS to the destination can be effectively achieved. In our work, this algorithm is developed in a CPN framework.

**IV. CPN MODEL FOR PATH SEARCH ALGORITHM**

**A. Hierarchical Coloured Petri nets**

**Coloured Petri nets (CPN):** is a graphical oriented language for design, specification, simulation and verification of systems. It associates color to each token distinguishing one token from the other. Different types of data manipulated in pipeline networks can be modeled based on this properties; for example alignment, state valve, pipeline network, etc. More in detail, and for example the pipeline network is modeled as a set of tokens, in a place Topology, each one containing the information \((pipe_1, valve, pipe_2)\) based on the rule: ”between two pipes there is one and only one valve”. The order of the pipes informs us the direction of oil flow. This allows the proposed model to be sufficiently generic to present any pipeline network independently of its size and its shape, and is also flexible for the modification of the pipeline network.

Furthermore, CPN is extended, firstly, with special types of arcs, the reset arc and the inhibitor arc; the first one does not impose a precondition on firing and empties the place when the transition fires and the second one imposes the precondition that the transition may only fire when the place connected to it has zero tokens residing in it. Secondly, priorities added to transitions, whereby a transition cannot fire, if a higher-priority transition is enabled (i.e. can fire).

**Hierarchical Petri nets:** CPN models can be hierarchically organised into a set of submodels small enough to be easily tracked. This facility allows the construction of a large model as a set of smaller models connected to each other using well-defined interfaces (substitution transitions and fusion places). The obtained global model will be more readable where submodels can be modified or changed independently of the upper module. Abstracted model will be made by a subset of a substitution transitions. A submodel will be associated with each substitution transitions to model its achievement.

To illustrate our approach, we used the previous study case which is developed using the the environment CPN Tools 4 [3] for editing, simulating, and analyzing CPN.

**B. The CPN model**

This section describes the CPN modeling technique used to develop Dijkstra’s algorithm for an oil-exporting seaport. In particular, the model developed in this paper is referred to the previous example (2), but it can be easily applied, with few modifications, to any generic pipeline network.

The model in Fig. 4 presents the upper layer of the hierarchical CPN model describing the generic behavior of the Dijkstra’s algorithm and each substitution transition models a specific step of the adopted algorithm.
- At the beginning, a token of the place "Path Tree" defines the order: source, destination and path reliability which is initialised to 1;
- the sequentiality of the three steps of Dijkstra’s algorithm is verified via a token in the place "Control Steps";
- When search is started, the place "Path Tree" contains the produced candidate paths. We have two cases: destination reached:
  - when the transition "End Of Search", which is a higher-priority one, is fired, the shortest reliable path is stored in the place "Path Found" with its shortest reliable path value.
destination not reached:
  - the substitution transition "Find Current Path" sets the path with the lowest path reliability in the place "Current Path";
  - the topology of the pipeline network is defined as a set of tokens in the place "Topology". Note that this simple technique makes the model almost independence of the network topology dimension and its eventual evolution;
  - the substitution transition "Determine Next Pipes" determines all unvisited neighbor pipes of the current path from the place "Topology" taking into consideration the pumps existence;
  - the paths found are stored in the place "New Paths";
  - the substitution transition "Calculi Paths Reliability" computes the $R_p$ of each new path.

1) Submodel "Find Current Path": in this CPN submodel (Fig. 5), the steps of the shortest path selection are:
- from "Path Tree", one path is chosen randomly;
- for each other path, the submodel compares the $R_p$ of the current path with that of the chosen path and make as chosen the path with the lowest $R_p$;
- when all candidate paths are tested, the lowest one is putted in "Path Find" and all the others are placed in "Paths Treated";
- before moving to the next step by firing the transition "End Finding", the transition "Recovering Other Paths" is fired enough times to recover all paths from the place "Paths Treated" to the place "Path Tree".

2) Submodel "Determine neighbor Pipes": this HLP-nets submodel (Fig. 6) aims to find all unvisited neighbor pipes of the chosen path.
If there are neighbor pipes: the transition "Finding the neighbor pipes" will be fired causing
  - the search of all unvisited neighbor pipes of the chosen path;
  - the creation of new paths, for each pipe found, as their concatenation with the current path;
  - for all pipes found, storage the tokens representing their connections of the other direction in "Visited Sub-tophology".
If there are no neighbor pipes: the transition "No next pipes" will be fired causing the deletion of the token representing the current path and the return to the previous step for finding another path.
3) Submodel "Calculi Paths Reliability": the main function of this CPN submodel (Fig. 7) is, for every new path created, the calculi of its $R_p$ (1). Therefore this submodel determines for each new pipe its opener and closers and incorporates a computation procedure to assess the dynamic reliability (2) of the determined valves using a place based on external events called "State Valve" which represents the characteristics of each elementary valve such as the initial instant of functioning, the identity, etc, in addition to its influence factors values: the commutation stress ($C_s$), the operating time ($T$), and the last valve state ($S$).

To achieve our goal, the following steps are achieved for every new path:

- determination of the opener which is situated between the two last pipes in the path and the computing of its dynamic reliability via the transition "Opener Computing Reliability";
- determination of their corresponding closers from all the remaining tokens in the place "Selected Sub-topology" with the computation of their dynamic reliability and updating the $R_p$ via the transition "Closers Computing Reliability";
- placing the used sub-topologies in the place "Sub-topology";
- if there are still pipes: the transition "Recovering Pipes" is fired enough times to recuperate all tokens in the place "Selected sub-topology" from the place "Sub-topology". These tokens represent all neighbor pipes of the current path;
- if there are no more pipes: the transition "Pipes Treated" is fired enough times to place all tokens from the place "Sub-topology" to the place "Visited Sub-topology";
- at the end, the transition "End Treatment" is fired in order to return to the first step.

V. THE CASE STUDY: RESULTS AND SIMULATION

A. Results

Firstly, to extract the coefficients of the failure rate base function and of the influence function, we have considered the OREDA database[12].

It indicates, the valve’s average lifetime is equal to 98, 500 hours (i.e. more than 11 years). So, we divided this lifetime between the two considered phases:

- useful life phase ($\beta = 1$) for the first 71, 600 hours (i.e. more than 8 years): it’s equivalent to the operational time of these type’s valves estimated by OREDA based on data collector.
- wear-out phase ($\beta = 4.748$) for the rest.

The burn-in phase (or infant mortality) is not treated because it’s not included in the OREDA database: it’s assumed that the data collection is started with the useful life phase. The $\eta$ determined is 98522.167 hours; and the calculated coefficients of influence factors are listed in Table I. Secondly, the proposed CPN model can be connected to the real pipeline to be supervised. Thus, the tokens in the place “State Valve” will model at any time the current state of the valves and will indicate precisely the values of their associated influence factors ($C_s, T, S$). Based on that information, the dynamic reliability value associated with each valve is obtained.

To illustrate our example (Fig. 2), let us assume that the pipeline has already been used to satisfy previous orders. Thus, each valve is characterized by a set of its influence factors modified throughout its functioning. Using the previous results, the current valves’ state is initialized with the values of table II.

Finally, the CPN model of the Dijkstra’s algorithm represents the exploitation of the openers and the closers states to achieve the most reliable alignment. As application, we will use the pipeline network example of Fig. 2 and Fig. 3 to answer to the request of transporting the oil from tank $T_1$ to the loading arm $L A_3$.

To reach the destination point $L A_3$ from tank $R_1$, the developed algorithm exploits iteratively openers and closers associated with the candidate paths to to compute $R_p$. As a result, the most reliable path computed by our model is $T_1, C_1, C_5, C_8, L A_3$ and $R_p = 0.32610$.

When an alignment is chosen to satisfy one or more requests, it will be used throughout the period necessary for filling the...
tanker. If this period is too long it would impact the dynamic reliability of its openers and closers. This impact will favor another alignment in the future to satisfy a similar request.

B. Simulation

During the simulation of the shortest path search, the designer can particularly observe the evolution of Dijkstra’s algorithm steps: selection of current path, determination of neighbor pipes and calculi of \( R_p \). A set of monitors can be integrated into the CPN model to estimate, based on dynamic reliability, valves’ lifetime and their maintenance timeout. Furthermore, the simulation allows the validation of certain properties of the studied system:

- **The deadlock-free states**: a deadlock is a situation wherein two or more components are waiting for the other to finish. We have easily verified that the state spaces of the studied example do not include deadlock states.
- **Flexibility**: the modification of pipeline network (i.e. maintenance, extension, etc.) does not influence the developed model because the related information of the network is specified only in the tokens.
- **Boundedness**: the number of paths in the place “PathTree” which represents the determined reliable paths, is bounded because it is less than or equal to the number of all possible paths in the network which is also bounded.

**VI. CONCLUSIONS AND PERSPECTIVES**

This research work addressed the problem of determining dynamic shortest path in oil pipeline network where reliability varies with time and with conditions of use. The study proposed, firstly, the definition of dynamic reliability and the determination of its parameters; secondly, the application of a Dijkstra’s algorithm for determining the most reliable path from the dynamic reliability assessment of openers and closers based on a CPN model.

The optimization of transportation networks based only on reliability may be considered a rather limited approach, since reliability is one facet of the problem. Other aspects, such as speed of transportation, equilibrated use of resources, etc., should be considered in a more hollistic approach to the problem. This suggests the use of multicriteria approaches, that will be the object of our attention.

**ACKNOWLEDGMENT**

The authors are grateful to Eric Niel for his helpful comments.

**REFERENCES**


\[ \begin{array}{|c|c|c|c|} \hline \text{Valve} & \text{last state} & C_v & t(\text{hour}) \times 10^4 \text{ (hour)} \\
\hline 1 & \text{close} & 451 & 20848 \times 10^4 \\
2 & \text{close} & 169 & 8012 \times 10^4 \\
3 & \text{close} & 282 & 11768 \times 10^4 \\
4 & \text{open} & 423 & 20176 \times 10^4 \\
5 & \text{close} & 401 & 19148 \times 10^4 \\
6 & \text{open} & 376 & 18024 \times 10^4 \\
7 & \text{open} & 476 & 22824 \times 10^4 \\
8 & \text{open} & 308 & 14592 \times 10^4 \\
9 & \text{open} & 242 & 11608 \times 10^4 \\
10 & \text{open} & 349 & 16700 \times 10^4 \\
11 & \text{open} & 269 & 12880 \times 10^4 \\
12 & \text{close} & 357 & 17092 \times 10^4 \\
13 & \text{close} & 327 & 15672 \times 10^4 \\
14 & \text{open} & 188 & 8912 \times 10^4 \\
15 & \text{close} & 248 & 11852 \times 10^4 \\
16 & \text{close} & 222 & 10628 \times 10^4 \\
17 & \text{close} & 500 & 23000 \times 10^4 \\
18 & \text{close} & 499 & 23952 \times 10^4 \\
\hline \end{array} \]