**ANITA GUDELJ**, Ph.D.
E-mail: anita@pfst.hr
**DANKO KEZIĆ**, Ph.D.
E-mail: danko@pfst.hr
Faculty of Maritime Studies, University of Split
Zrinsko-Frankopanska 38, 21000 Split, Croatia
**STJEPAN VIDAČIĆ**, Ph.D.
E-mail: svidacic@foi.hr
University of Zagreb,
Faculty of Organization and Informatics
Pavlinska 2, 42000 Varaždin, Croatia

Science in Traffic and Transport
Original Scientific Paper
Accepted: Nov. 27, 2011
Approved: Nov. 13, 2012

# MARINE TRAFFIC OPTIMIZATION USING PETRI NET AND GENETIC ALGORITHM

***ABSTRACT***

*The paper deals with the traffic control and job optimization in the marine canal system. The moving of vessels can be described as a set of discrete events and states. Some of these states can be undesirable such as conflicts and deadlocks. It is necessary to apply adequate control policy to avoid deadlocks and blocks the vessels' moving only in the case of dangerous situation. This paper addresses the use of Petri net as modelling and scheduling tool in this context. To find better solutions the authors propose the integration of Petri net with a genetic algorithm. Also, a matrix based formal method is proposed for analyzing discrete event dynamic system (DEDS). The algorithm is developed to deal with multi-project, multi-constrained scheduling problem with shared resources. It is verified by a computer simulation using MATLAB environment.*

***KEYWORDS***

*marine traffic system, optimization, deadlock avoidance, genetic algorithm, Petri net*

## 1. INTRODUCTION

A canal system enables the connection of two water surfaces that have different levels of water. The vessels passing through the canal system in opposite directions use various resources (canals, locks, basins). Some of these resources are non-shared (resource that can be occupied by the vessels moving in only one direction), and others are shared resources. As a rule the number of vessels in the resources is limited. The main problems that have to be solved are to [1]: a) control traffic in a way to ensure safe passage through the canal system, b) resolve possible conflicts in case that more vessels require shared resource in the same time, c) avoid possible deadlocks, d) ensure the vessel passage in the shortest time possible.

The marine canal traffic systems (MCTS) can be generally described as DEDS [2]. In our preliminary study [3], the authors used DEDS and Petri net (PN) theory, a well-known tool for analyzing DEDS, to resolve some of the above mentioned problems. Also, for MCTS, the crucial management problem is optimizing the balance between the vessel-owners who request quick vessel movements and economical use of resources. It is desirable to utilize both vessels and resources as intensively as possible. So, job scheduling is a very important issue in the planning and operation of MCTS. The problem of scheduling jobs bounded by resources constraints, resources availability and precedence relations is an exceedingly difficult task. It is especially true if an attempt is made simultaneously to minimize total completion times of all jobs to precedence relations between the jobs and the limited renewable resource availabilities, and meet conflict and deadlock avoidance criteria. This problem type is known to be NP-hard [4]. In the last decade, genetic algorithms (GA) have been applied to many optimization problems, including scheduling [5]. The reason is that GA is known to search efficiently a large search space, without explicitly requiring additional information (such as availability of derivative information) about the objective function to be optimized [6].

There are many published papers on planning and scheduling. Many authors have tried to solve the deadlock problem by PN. Ezpeleta developed an algorithm for deadlock prevention for the ordinary and conservative S3PR [7] class of PNs. Lautenbach [8] investigated the algorithm for finding the minimal siphons inside the net as well as the algorithm for deadlock prevention by control places for ordinary PNs which do not contain source places. Lewis [9] developed an efficient algorithm for deadlock prevention in the specific class of PNs that describes FMS. Resource-constrained project scheduling has been a research topic in the last 3 decades, resulting in a wide variety of optimization procedures. Moreover, due to the absence of real-time and accurate production data, most of the developed methodologies cannot be used in real-life production control.

The goal of this paper is to find optimal, conflict and deadlock free schedules in MCTS by an algorithm which combines the search method of GA with PN structural analysis procedure as a criterion to make the job scheduling optimal and deadlock free. Modelling marine traffic system is the first task to achieve this. The class of MRF1PN, subclass of flowline system PN (FPN) is used in this paper. FPN is a class of Petri net which is basically designed for analyzing finite buffer multi class re-entrant flowline systems (MRF) large class of flexible manufacturing systems (FMS) [3, 10]. MRF1PN contains some specific properties, which will be discussed in the next subsection.

The paper is organized as follows: Section 1 introduces briefly the Petri nets and genetic algorithms. Also, this section reviews the basics of P/T Petri net properties and describes also the notations which are used throughout the paper. In this section special attention is dedicated to MRF1PN class. Section 2 describes our approach to how to solve the scheduling problem and deals with integration PN and GA which we proposed for the scheduling problem. Section 3 introduces marine traffic system as a case study. The moving of vessels and adequate control policy of the supervisor is verified using P-timed PN and GA. The concluding remarks are made in Section 4.

### 1.1 Petri nets and deadlock avoidance

PN is a graphical and mathematical modelling tool which can be used as a visual communication aid. Basically, PN is a bipartite graph consisting of two types of nodes, places and transitions, connected by arcs.

Petri net is a 6-tuple [11]:

$$PN = (P, T, \mathbf{I}, \mathbf{O}, w, m_0), \tag{1}$$

where $P$ is the set of places, $T$ is the set of transitions, $\mathbf{I}_{(n,m)}: T \times P \to \{0,1\}$ is an input incidence matrix, $\mathbf{O}_{(n,m)}: T \times P \to \{0,1\}$ is an output incidence matrix, $w: [\mathbf{I} \vdots \mathbf{O}] \mapsto \{1,2,3,\ldots\}$ is a weight function, $m_0$ is initial marking.

Places and transitions $v \in P \cup T$ are called nodes and denote states and events in the DES. In PN with $m$ places and $n$ transitions, the incidence matrix $\mathbf{W}$ is a $n \times m$ matrix whose elements are $a_{ij} = w(t_j, p_i) - w(t_i, p_j)$, where $w(t,p)$ is the weight of the arc between $p, t$. If all arcs in PN have weight equal to 1, it should be noted that

$$\mathbf{W} = \mathbf{O} - \mathbf{I}. \tag{2}$$

The matrices $\mathbf{I}$ (input matrix) and $\mathbf{O}$ (output matrix) provide a complete description of the structure of PN. If there are no self loops, the structure may be described by $\mathbf{W}$ only. The incidence matrix allows an algebraic description of the evolution of the marking of PN. The marking of PN changes from marking $m_k$ to marking $m_{k+1}$:

$$m_{k+1} = m_k + \mathbf{W}^T \cdot \sigma. \quad (3)$$

where $\sigma$ is a transition vector.

The transition vector $\sigma$ is composed of non-negative integers that correspond to the number of times a particular transition has been fired between markings $m_k$ and $m_{k+1}$.

Siphon S is the set of PN places for which it is true that each transition having an output arc from the set also has an input arc into the set ($\bullet S \subseteq S \bullet$). Trap T is the set of places for which it is true that each transition having an input arc into the set also has an output arc from the set ($T \bullet \subseteq \bullet T$). Once the trap becomes marked, it will always be marked for all future reachable markings. Once the siphon becomes empty, it will always remain empty. More about siphons and traps can be found in [9].

The flowline system Petri net (FPN) [12] is the subclass of P/N Petri net specially designed for analyzing MRF flexible manufacturing systems, where each part type $k$ is characterized by predetermined sequence of jobs $J^k = \{J_1^k, J_2^k, \ldots, J_{L_k}^k\}$, with at least one resource for each job. In the FPN, set of places P is given by $P = R \cup J \cup J_{in} \cup J_{out}$, as the set of resources $R$ plus the set of units arrivals $J_{in}$ plus the set of finished units $J_{out}$ and plus the set of places representing the ongoing jobs $J$. For any $r \in R$, define the job set $J(r)$ as the set of jobs using the resource $r$, and resource loop $L(r) = r \cup J(r)$. Given a subset of resources $Q \subset R$, define the job set of $Q$ as $J(Q) = \bigcup_{r \in Q} J(r)$. Denote $R(J_i^k)$ the resources used by job $J_i^k$.

The system described in section 2 belongs to the class of MRF1PN. MRF1PN guarantees that (i) there are no self loops, (ii) each unit-path has a well defined beginning and an end, (iii) every job requires one and only one resource with no two consequent jobs using the same resource, (iv) and (v), there are no choice jobs and no assembly jobs, (vi) there are shared resources. In MRF1PN, for any two $r_i, r_j \in R$, $r_i$ is said to wait $r_j$, denoted $r_i \to r_j$, if the availability or $r_j$ is an immediate requirement for the release or $r_j$, i.e., $\bullet r_i \cap r_j \bullet \neq \varnothing$[1].

Any set of resources is called circular wait C, if among the set of resources $r_a, r_b, \ldots, r_w$ there exist wait relations such that $r_a \to r_b \to \ldots \to r_w$ and $r_w \to r_a$. C relations are characteristic among shared and non-shared resources in MRF1PN and contain at least one shared resource.

From a circular wait C, a deadlock state called circular blocking CB can be reached. A circular wait C is said to be in CB if (i) $m(\mathrm{C}) = 0$; and (ii) for each $r \in \mathrm{C}, \forall p \in J(r)$ with $m(p) \neq 0$, $p\bullet \in \mathrm{C}\bullet$. Avoiding CB is necessary but generally not sufficient for deadlock-free dispatching policy. To prevent deadlock in MRF1PN we must first avoid CB conditions, which are closely related to the critical siphon. A critical siphon is a minimal siphon that does not contain any resource loop. The next step is to find sets of job places, so-called critical subsystems $J_0(C)$, which are very important in deadlock prevention.

C is in CB at any $m_0 \in \Re(m)$ if and only if siphon becomes empty. The siphon is empty if and only if $m(J_0(C)) = m_0(C)$. The token sum in the critical subsystem $J_0(C)$ must be limited above $m_0(C)-1$. To avoid such first level deadlocks we must connect control places to the transitions before and after any critical subsystems $J_0(C)$ which make sure that the token sum in the critical subsystem $J_0(C)$ is limited to $m_0(C)-1$.

It must be noted that when a system contains key resource, the system may run in the so-called cyclic circular wait relation and also run in the so-called second level deadlock several steps before any CB actually occurs. To avoid the second level deadlock we must find key resource and apply control policy to make sure that key resource does not remain the last available resource [12]. More can be found in [4, 13].

### 1.2 Genetic algorithm

Genetic algorithms (GA) are population-based heuristic approaches. The principles of GA were developed by Holland [14]. GA starts with a set of feasible solutions (population) and iteratively replaces the current population by a new population. The reproduction mechanism selects the parents and recombines them using a crossover operator to generate offspring. Reproduction and crossover operators tend to increase the quality of the populations and force convergence. Mutation replaces genetic material. GA is able to evolve solutions to real world problems, if they have been suitably encoded. It requires a fitness function that represents a measure of the quality of each encoded one.

## 2. NEW APPROACH OF PN AND GA INTEGRATION

In the area of optimization, there is a need for ongoing research concerning the integration of resource scheduling problems, so that the integration can be applied in practice. There are recent trends to combine these problems, even though this integration is neither easy nor intuitive [15, 16, 5].

In this paper a new model that integrates PN and GA for multi-project resource scheduling problems is proposed. The aim is to demonstrate the applicability of the integration of PN and GA in finding optimized job schedules with no conflicts and deadlocks in multi-project[2] system with multi-shared resources. The matrix approach of deadlock supervisor design, which was originally developed for re-entrant FMS, is selected [17, 9].

*Figure 1* describes the framework of this new model. MRF1PN is chosen as modelling tool because a matrix model can only be applied to this class of PN and can be involved in GA in a simple way. It provides the rigorous framework needed for the analysis and simulation of DES. To compute, at each timestamp *t*, which current jobs are completed and resource released or which resources are ready, P-timed PN is used.

Job and resource places are divided in two parts: input part *mi*, where tokens are positioned as the place input transitions fire; end part *mf*, where token is moved from the part *mi* after the time associated with that place has lapsed, and the token becomes available to fire the place output transitions. These associated times to the places request the subroutine for computing elapsed times, completed currently ongoing jobs and released resources. Now it is possible to compute different performance measures such as wasted times, delays, resource percent utilization and so on.

GA is responsible for evolving the chromosomes which are comprised of real random numbers between 0 and 1. Using matrices from MRF1PN, each chromosome is decoded in the priorities of the jobs, delay and release times. After that, the set of chromosomes shall be forwarded to the schedule generator. This procedure makes use the priorities, the delay and release times defined in the first phase and constructs parameterized schedules. Schedules are defined from generated firing sequence which has to be checked for conflict and deadlock existence. After a schedule is obtained, the corresponding measure of quality (minimum makespan) is a feedback to the genetic algorithm.

### 2.1 Chromosome representation

Two commonly used approaches for chromosome representation are direct and indirect encoding. In direct encoding, a chromosome completely represents a solution while in indirect encoding a chromosome contains data which are used to obtain a solution. During our research, we have used the indirect encoding, because the direct use of schedules as chromosomes was too complicated to represent and manipulate. To

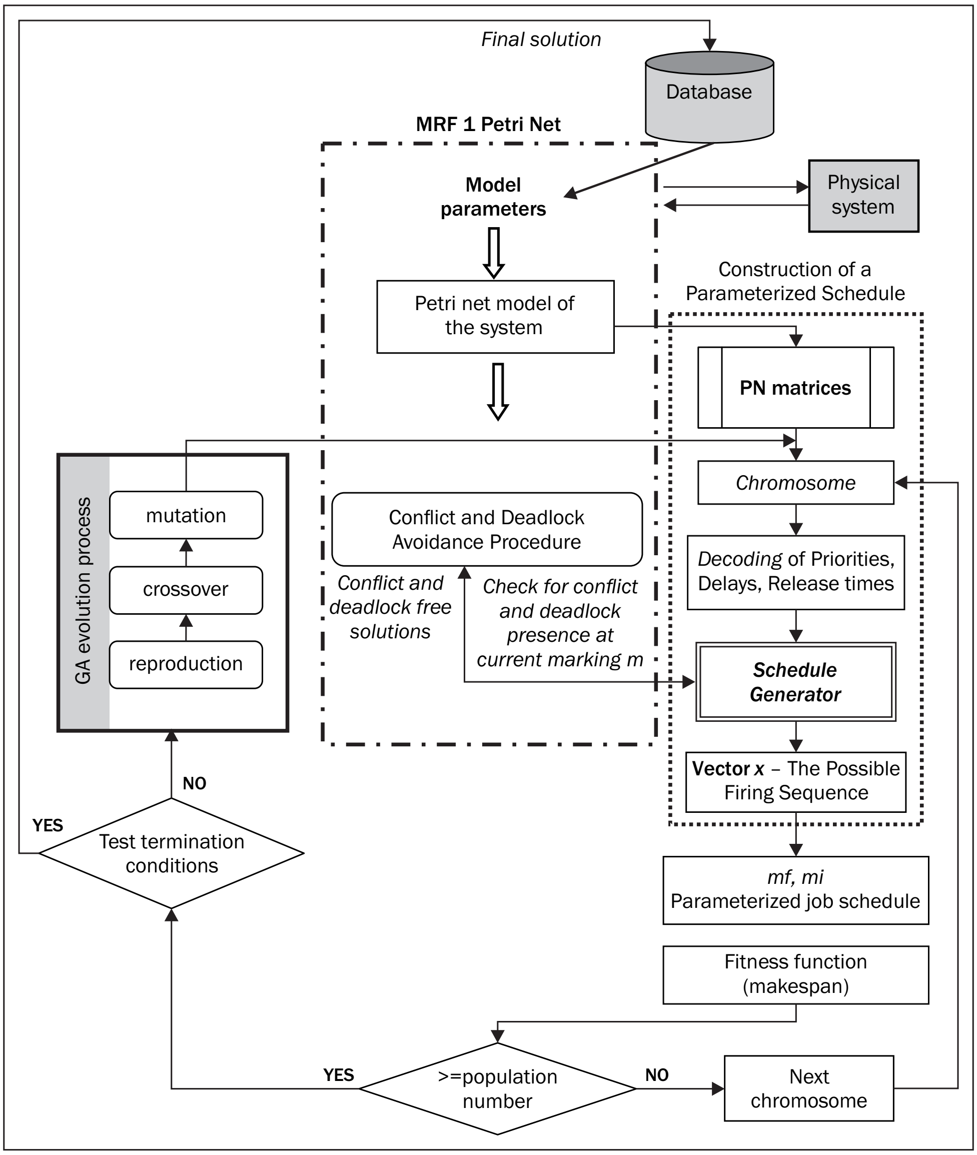
Final solution
Database
MRF 1 Petri Net
Model parameters
Physical system
Petri net model of the system
Construction of a Parameterized Schedule
PN matrices
Chromosome
GA evolution process
mutation
crossover
reproduction
Conflict and Deadlock Avoidance Procedure
Conflict and deadlock free solutions
Check for conflict and deadlock presence at current marking m
Decoding of Priorities, Delays, Release times
Schedule Generator
Vector x – The Possible Firing Sequence
NO
YES
Test termination conditions
mf, mi Parameterized job schedule
Fitness function (makespan)
YES
>=population number
NO
Next chromosome

*Figure 1 - A framework for integration of PN and GA*

obtain the feasible solution, the decoding procedure, also called schedule generator, must be used. In this work the extended parameterized active schedule generator is developed, which extends the algorithm proposed in [18]. Each chromosome is encoded as random string consisting of $2m+n+x$ real random numbers between 0 and 1, where $n$ is the number of jobs, $m$ the number of projects and $x$ the number of resources. Hence, a chromosome is divided into sections (4), and crossover and mutation can be performed only inside a section.

$$\Big(\underbrace{0.32, 0.77, \ldots, 0.52}_{m\text{ genes}}, \underbrace{0.53, \ldots, 0.66}_{n\text{ genes}}, \underbrace{0.42, \ldots, 0.68}_{m\text{ genes}}, \underbrace{0.91, \ldots, 0.85}_{x\text{ genes}}\Big) \qquad (4)$$

The first $m$ genes are used to determine the priorities of each project. The genes between $m+1$ and $m+n$ are used to determine the delay time used at each of the $n$ iterations of scheduling procedure which schedules one job per iteration. For this problem, the delay times are generated by

$$DelayGen_g = gene_g \times MaxDur \times 1.5, \qquad (5)$$

where $MaxDur$ is the maximum duration among all job durations, $g = 1, \ldots, n$.

The genes between $m+n+1$ and $2m+n$ are used to determine the release time of each project. The times are determined by

$$rt_i = ERD_i + gene_{m+n+i} \times (DD_i - ERD_i), \qquad (6)$$

where $ERD_i$ represents the earliest release date for project, $DD_i$ represents due date for each project $i = 1, \ldots, m$.

The last $x$ genes are used to determine the release dates of each resource:

$$RT_i = gene_{2m+n+i}, \ \forall i = 1, \ldots, x. \qquad (7)$$

## 2.2 Schedule generation procedure

The schedule generation procedure is used to construct schedules which are based on a partial job scheduling that does time incrementing and the description of the procedure shown in *Figure 2*.

This procedure constructs active schedules (schedules in which no operation could be started earlier without delaying some other operation or breaking a precedence constraint). An optimal schedule is always active, so the search space can be safely limited to the set of all active schedules. But, the set of active schedules contains many schedules with relatively large delay times, and therefore poor quality in terms of makespan. In order to reduce the solution space and to control the delay times, we used the concept of parameterized active schedules introduced in [18].

As input parameters the procedure uses matrices $F_r$, $S_v$ from matrix description of MRF1PN [12]. Time $t(g)$ associated to the iteration $g$ is the time when some jobs are scheduled. The set $J$ contains all jobs which have not been scheduled up to time $t(g)$. Let $A(t_g)$ be set of all jobs which are active at $t(g)$. It is easy to determine this set by the input vector *mi* during timed simulation. If some element from the vector *mi*, in the iteration *i* of timed simulation, equals 1 then a specific job is active in that timestamp.

While being processed, job $j$ requires resource type $k$ during every time instant of its non-preemptable duration $d_j$. Resource type $k$ has a limited availability $RD(k,:)$ at any point of time. $S_g$ comprises all jobs which have been scheduled up to iteration $g$, and $F_g$ comprises the finish times of the jobs in $S_g$.

### 2.2.1 Evolutionary process

In this work, reproduction is accomplished by first copying 20% of the best individuals from one generation to the next, in what is called an elitist strategy (Goldberg, 1989). The advantage of an elitist strategy over traditional probabilistic reproduction is that the best solution is monotonically improving from one generation to the next. The potential downside is population convergence to a local minimum. This can, however, be overcome by high mutation rates. The mutation

```
% Input parameters from MRF1PN:
    Fr – the resource_requirements matrix; Sv – job_start matrix; mi – input part; mf – end part; PNtimes – the timed vector;
% Input parameters from GA:
    Ch – decoded chromosome; DL – the part of the chromosome which determines Delays; r – the earliest start time for each job calculated from Ch;
% User defined parameters for timed simulation:
    TotalTime – total simulation time; TimePeriod – time period for evaluated the marking of PN; Num_iter=TotalTime/TimePerid – total number of iterations;
% Results:
    Ft – matrix of job start times (time of firing transitions) for optimal schedule; S0 – optimal schedule;
Algorithm
1.  Initialization:
    g=1; A=[]; S0=[0]; % S0 involves initial dummy job
    E=[]; % The set of all jobs which are precedence feasible in the schedule iteration g
    FMC=[]; % Matrix of earliest finish time for scheduled jobs
    t(1)=min(DL); RD(k,1)=R(k); %R-initial vector of resource capacities for each resource
2.  Timed simulation:
    for each iteration from 1 to Num_iter do
3.  Loop 1: while ($J \neq \varnothing$) do
    Determine the set E;
4.  Loop 2: while ($E \neq \varnothing$) do
    a)  Determine the set $S_g$ – jobs from E which satisfy the precedence constraints and constraints in term of limited resource capacities;
    b)  For jobs from $S_g$ calculate $x$ – transition firing vector for MRF1PN;
    c)  Conflict and deadlock avoidance procedure: If there are conflicts and deadlocks, calculate new vector $x$;
    d)  Calculate the marking of PN for each time unit during iteration $g$;
5.  Calculate the earliest finish time for jobs from $S_g$:,
    $FMC(g) = \min_{j^* \in S_g}(F_g(j^*))$ if $r_{j^*,k} \leq RD(k,\tau); \tau \in [t_g, t_g + d_{j^*}]$    for each resource $k$
6.  Update: $S0 = S0 \cup S_g, Ft = Ft \cup FMC(g)$
7.  Iteration increment: g = g + 1;
8.  Update: A(t), resource capacities RD(k,t) for $\forall t \in [Ft(g), FMC(g)]$;
9.  End of loop 2
10. $t(g) = \min(t \in Ft : t > t(g))$ %Determine the time associated with iteration g;
11. End of loop 1;
12. End of time simulation;
13. Calculate makespan; %end of procedure
```

*Figure 2 - Pseudo-code of schedule generation procedure*

consists of replacing 30% of the worst parents of the existing population by new generated chromosomes. The rest of the new population is created by the crossover operator which consists of exchanging the information of two parents of the existing population to produce two new chromosomes.

## 3. CASE STUDY

### 3.1 Marine traffic system

*Figure 3* presents an example of MCTS as a simple MRF system constrained by resource capacities.

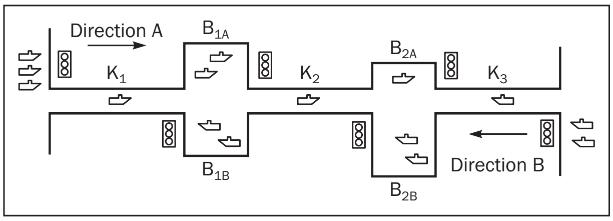

*Figure 3 - Marine canal traffic system*

This system consists of three canals $K_1, K_2, K_3$ and four basins $B_{1A}, B_{2A}, B_{1B}, B_{2B}$. All canals and basins represent resources of a canal system. The vessels at the left end of the canal system wait for the passage to the right in the direction of A, and vessels at the right side wait for the passage to the left in the direction of B.

The vessel in direction A must pass $K_1 \rightarrow B_{1A} \rightarrow K_2 \rightarrow B_{2A} \rightarrow K_3$, and the vessel in direction B must pass $K_3 \rightarrow B_{2B} \rightarrow K_2 \rightarrow B_{1B} \rightarrow K_1$.

The vessels in both directions share canals. The basins are designed only for one direction and for waiting for the availability of the next canal. If a particular resource is occupied in a moment of time, and if there are vessels waiting to use them, then these vessels wait for the availability of the occupied resource at the exit of the resource where they are in the moment of time. When the resource becomes available, it is occupied by the awaiting vessels. The moving of the vessels through the system is limited due to the capacity of resources. The canal capacities are $K_1, K_2, K_3 = 1$, the basin capacities are $B_{1A} = 3, B_{2A} = 1, B_{1B} = B_{2B} = 2$ vessels.

The traffic in the canal system can be fully controlled using the computer system which controls traffic lights at the entrance into resource in directions A and B. The sensors for detection of vessels passing from one resource to the other are connected to the computer control system. The traffic light signalization system should not let vessels enter a resource in order to avoid the forbidden states.

The intention of this section is to clarify the theory in the previous section and the proposed algorithm of integration MRF1PN and GA will be applied on the example of MCTS. Although this system is a simple one, the method extends directly to complex systems. The first step is to make MRF1PN model of the canal system (*Figure 4*).

The tokens in input places $\{p_1, p_2\}$ represent the vessels waiting for entering the system. The set of output places is $\{p_{26}, p_{27}\}$. The set of all places that represent jobs in the system (the number of tokens in a job place represents the number of vessels in particular resource) is $\{p_3 - p_{12}\}$, and the set of places that represent the availability of resources is $\{p_{13} - p_{19}\}$. Control places $\{p_{20} - p_{25}\}$ belong to the supervisor which ensures the conflict and deadlock free operation of the net by adding or removing tokens from these places. Initially, the number of tokens in each control place equals 1.

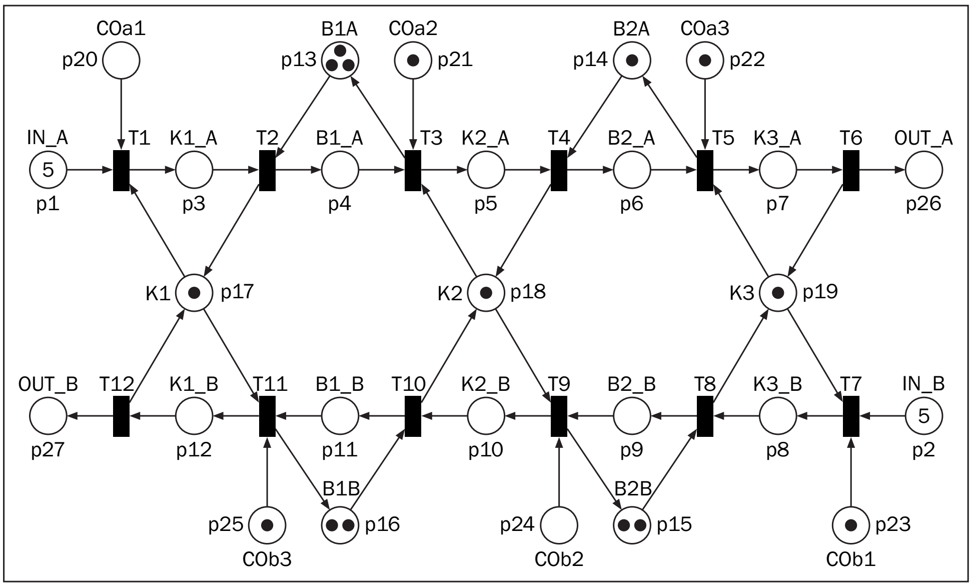

*Figure 4 - Petri net model of marine canal traffic system*

### 3.2 Timed simulation

As distinguished from many authors, we take in account the dynamic behaviour of marine traffic system. In reality, any real traffic system has nonzero times for performing jobs and setting up resources once they are released. In our PN, these times are interpreted as timed places for jobs and resources and they are defined by following *PNtimes* vector

$$PNtimes = \Big[\overbrace{0,0,0.54,1,1.08,1,0.81}^{\text{job times in direction A}},$$
$$\overbrace{0.81,1,1.08,1,0.54}^{\text{job times in direction B}},$$
$$\overbrace{0.9,0.3,0.6,0.6,0.02,0.02,0.02}^{\text{resource setup times}},$$
$$0,0,0,0,0,0,0,0,0\Big] \qquad (8)$$

Assume that vessel speeds through the canals are limited to one knot (v =1.852km/h) and the canal lengths are as follows s1=1km, s2=2km, s3=1.5km. These times are associated to places $p_3, p_5, p_7$ for vessel movement in direction A, and to places $p_8, p_{10}, p_{12}$ for direction B:

$$PNtimes(3) = PNtimes(12) = s_1/v = 0.54,$$
$$PNtimes(5) = PNtimes(10) = 1.08$$
$$PNtimes(7) = PNtimes(8) = 0.81.$$

Also, the assumption is that every vessel must remain in the basins B1A_A = B2A_A = B2B_B = B1B_B = 1. Set up times for resources are associated to places $\{p_{13} - p_{19}\}$.

### 3.3 Conflict and deadlock resolution

Conflicts arise when vessels from both directions try to occupy the same shared resources $\{p_{17} - p_{19}\}$. In this situation the transitions $t_1, t_{11}$ and $t_3, t_9$ and $t_5, t_7$ are in conflict (both transitions are enabled at the same time). A conflict-free supervisor enables only one direction (direction A or direction B) with equal probability. The second problem is how to produce the design of a deadlock-free supervisor. To achieve this we must apply the matrix approach described in [9, 17]. Since these calculated matrices are too large to be presented here, only the results are presented here.

**Step 1**: There are 7 P-invariants in the net:

$$P_1 = \{p_4, p_{13}\},\ P_2 = \{p_6, p_{14}\},\ P_3 = \{p_9, p_{15}\},$$
$$P_4 = \{p_{11}, p_{16}\},\ P_5 = \{p_3, p_{12}, p_{17}\},\ P_6 = \{p_5, p_{10}, p_{18}\},$$
$$P_7 = \{p_7, p_8, p_{19}\}.$$

**Step 2**: There are 2 SSW

$C_1 = \{p_{13}, p_{16}, p_{17}, p_{18}\}$, $C_2 = \{p_{14}, p_{15}, p_{18}, p_{19}\}$, and 1 CCW $C_3 = C_1 + C_2 = \{p_{13}, p_{14}, p_{15}, p_{16}, p_{17}, p_{18}, p_{19}\}$.

**Step 3**: There are 3 critical siphons:

$$S_{C_1} = \{p_5, p_{12}, p_{13}, p_{16}, p_{17}, p_{18}\},$$
$$S_{C_2} = \{p_7, p_{10}, p_{14}, p_{15}, p_{18}, p_{19}\},$$
$$S_{C_3} = \{p_7, p_{12}, p_{13}, p_{14}, p_{15}, p_{16}, p_{17}, p_{18}, p_{19}\}$$

and 3 critical subsystems

$$J_{0_1} = \{p_3, p_4, p_{10}, p_{11}\},\ J_{0_2} = \{p_5, p_6, p_8, p_9\},$$
$$J_{0_3} = \{p_3, p_4, p_5, p_6, p_8, p_9, p_{10}, p_{11}\}.$$

**Step 4**: The initial marking of $C_1, C_2, C_3$ are $m_0(C_1) = 6$, $m_0(C_2) = 5$, and $m_0(C_3) = 11$.
To avoid the first level deadlock the number of vessels in critical subsystems must be limited by:

$$m_0(J_{0_1}) \leq 5,\ m_0(J_{0_2}) \leq 4,\ m_0(J_{0_3}) \leq 10. \qquad (9)$$

**Step 5**: This system is irregular because key resource exists:

$$C_1 \cap C_2 = \{K2\} = \{p_{18}\},\ m_0(p_{18}) = 1. \qquad (10)$$

**Step 6**: To ensure the absence of the second level deadlock the key resource $K2$ must not be the last available resource in the system.

The deadlock prevention supervisor which applies the control policy derived in steps 4 and 6 is verified using computer simulation of vessels moving through the canal system. We are simulating the process of moving five vessels in direction A, and five vessels in direction B simultaneously. In case of conflict, random dispatching is used.

*Figure 5* shows markings of input and output places and it is possible to see that the last vessel leaves the canal system 17.08 time units from beginning of vessels moving through the canal system. This is the shortest possible time for the whole process, and applied control policy ensures no conflicts and no deadlocks in the system.

The algorithm of integration PN & GA was applied on the same MCTS, as described in the previous section. The objective is to find a project schedule so that the vessels pass through the canals in the shortest time possible, i.e. to minimize the makespan (*Fmax*) as the time when the last vessel leaves the canal system.

The algorithm is tested ten times with the following configuration:

*Number of jobs* = 50,
*Population size* = 0.2 * *Number of jobs*,
*End condition* = 20 *generations for GA*,
*Selection = elitist*,
*Selection pressure* = 0.05,
*Probability of the parameterized uniform crossover* = = 0.75,
*Total simulation time* = 30,
*Time period* = 0.02.

*Figure 6* shows the results for the best schedule. The last vessel leaves the canal system after 15.18 timestamps.

The optimized schedule reduces the total time by 1.9 timestamps or approximately 11%. This is not a negligible saving for the ship owners, passengers and port agents because they want the vessels to pass

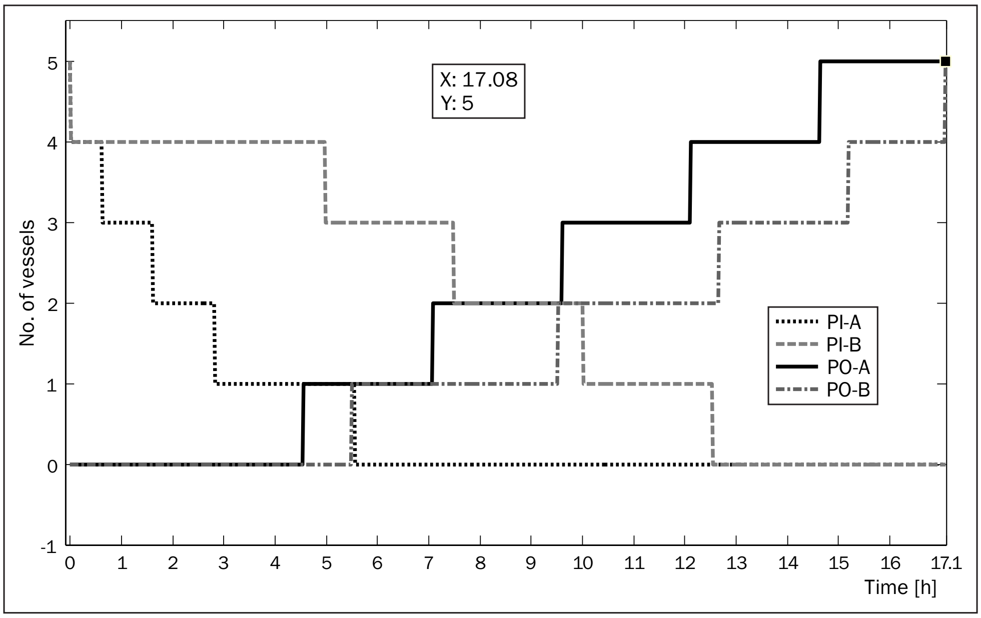

*Figure 5 - Number of vessels in input and output places*

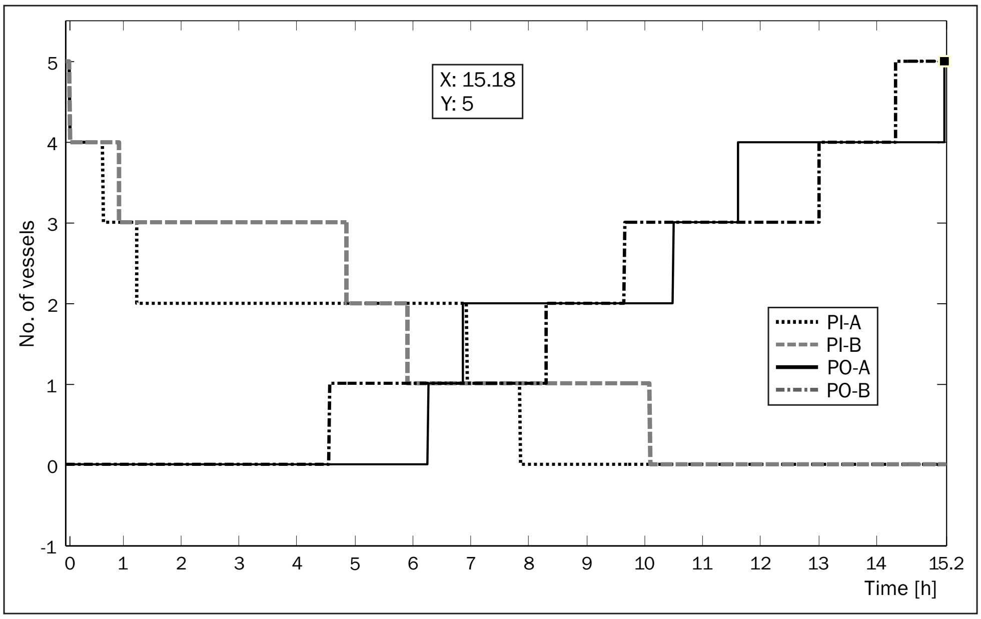

*Figure 6 - Optimized schedule; number of vessels in input and output places*

through the canals in the shortest time possible. If the number of vessels is increased, such as 10 vessels on each side of the system, and if optimization methods are not used, it will take about 40 hours [3]. With this integration PN and GA saving would be about 4.35 hours. For a longer time period this means big savings in time, in fuel consumption (if vessels wait with engines included) and effective resource usability.

## 4. CONCLUSION

Time resource management is important for any job scheduling problem. In this paper, new algorithm which integrates PN and GA is proposed to model a kind of schedule problem constrained by resources. To find an improved better solution in terms of performance measure in a complex search space, an algorithm which integrates P-timed MRF1PN and GA, including the matrix model for conflict and deadlock avoidance, is proposed. It is the main difference from other algorithms that combine PN and GA. The main contributions of this paper are as follows.

First of all, the proposed algorithm obtains parameterized active and feasible schedules with small computational efforts. Chromosomes are represented indirectly by parameters that are later used by a schedule generator to obtain a solution. Using this indirect representation of schedules as chromosomes it is possible to find optimal job schedule in any complex system contending for limited resources. Secondly, using the MRF1PN the major constraints in scheduling problems can be formulated with the matrix model. Therefore there is no need to mathematically define any constraints, variables or supervisor which results in a substantial reduction of problem formulation complexity.

Thirdly, to keep the atomic properties of any real production system, the jobs and resources are represented by places. To represent the times for performing jobs and setting up resources, times are associated to job and resource places in a way which is deferent from the traditional modelling approach which combines PN and GA. Each place is divided into two parts of the input and output of the associated time into which they were separated, and thus the power of the model is increased at the expense of slightly larger memory space occupied and a slightly slower performance program.

Dr. sc. **ANITA GUDELJ**
E-mail: anita@pfst.hr
Dr. sc. **DANKO KEZIĆ**
E-mail: danko@pfst.hr
Pomorski fakultet, Sveučilište u Splitu
Zrinsko-Frankopanska 38, 21000 Split, Hrvatska
Dr. sc. **STJEPAN VIDAČIĆ**
E-mail: svidacic@foi.hr
Sveučilište u Zagrebu,
Fakultet organizacije i informatike
Pavlinska 2, 42000 Varaždin, Hrvatska

## *SAŽETAK*

## *OPTIMIZACIJA POMORSKOG PROMETA POMOĆU PETRIJEVIH MREŽA I GENETSKOG ALGORITMA*

*Ovaj se rad bavi upravljanjem prometom i optimizacijom rasporeda poslova u sustavu prometa morskim kanalima. Kretanje plovila može se opisati kao sustav diskretnih događaja i stanja. Nužno je primijeniti odgovarajuću strategiju upravljanja kako bi se izbjegli konflikti i blokirati prolaz plovila u opasnim situacijama. U tom kontekstu, u radu se koristi Petrijeva mreža kao alat za modeliranje i određivanje rasporeda. Kako bi se pronašlo bolje rješenje, autori predlažu integraciju MRF1 klasu Petrijevih mreža (MRF1PN) i genetskog algoritma. Također, predlaže se i integracija matrične metode za analizu dinamičkog sustava s diskretnim događajem (DEDS). Algoritam je načinjen tako da radi s više projekata, s više-kriterijskim problemom rasporeda s višeradnim resursima. Testiran je pomoću računalne simulacije koristeći program MATLAB.*

## *KLJUČNE RIJEČI*

*pomorski prometni sustav, genetski algoritam, izbjegavanje zastoja, optimizacija, Petrijeva mreža,*

## REFERENCES

1. $\bullet t (t \bullet)$ is a set of input (output) places to transition $t$.
2. Each project is composed of $i$ jobs $J = \{N_1, \dots, N_i\}$, where jobs $N_1$ and $N_i$ are dummy and represent the initial and final jobs of project. $J$ is the set of jobs.

## LITERATURE

[1] **Degre, T.**: *The Management of Marine Traffic: A Survey of Current Future Measures*, Journal of Navigation, Vol. 48, 1995, pp. 53-69

[2] **Kezić, D.**, **Perić, N.**, **Petrović, I.**: *A Petri Net Approach of Deadlock Prevention in Marine Traffic System*, In Proceeding of the IEEE International Symposium on Industrial Electronics, Croatia, Vol. 1, 2005, pp. 217–222

[3] **Kezić, D.**, **Gudelj, A.**: *Design of River System Deadlock Avoidance Supervisor By Using Petri Net*, PROMET, Vol. 22, No. 3, 2010, pp. 215-221

[4] **Blazewicz, J.**, et al.: *Scheduling Subject to Resource Constraints: Classification and Complexity*, Discrete Applied Mathematics, Vol. 5, 1983, pp. 11-24

[5] **Kofjač, D.**, **Kljajić, M.**: *Application of Genetic Algorithms and Visual Simulation in a Real-Case Production Optimization*, Transactions on System and Control, Vol. 3, No. 12, 2008, pp. 992-1001.

[6] **Yoshikawa, M.**, **Terai, H.**: *Genetic Algorithm Engine for Scheduling Problems*, WSEAS Transactions on Circuits and Systems, Vol. 5, No. 3, 2006, pp. 397-402

[7] **Ezpeleta, J.**, et al.: *A Petri Net Based Deadlock Prevention Policy for Flexible Manufacturing Systems*, IEEE Trans. on Robotics and Automation, Vol. 11, No. 2, 1995, pp. 173-184

[8] **Lautenbach, L.**, **Ridder, H.**: *The Linear Algebra of Deadlock Avoidance – A Petri Net Approach*, Research Report of Institute for Computer Science, Germany, 1996.

[9] **Lewis, F.**, et al.: *Analysis of Deadlocks and Circular Waits Using A Matrix Model for Discrete Event Systems*, Automatica, Vol. 34, No. 9, 1998, pp. 0-19

[10] **Tricas, F.**, et al.: *An Iterative Method For Deadlock Prevention In FMS*, in Proceeding of 5th Workshop Discrete Event System, Belgium, 2000, pp. 139 – 148

[11] **Murata, T.**: *Petri Nets: Properties, Analysis, and Applications*, In Proceedings of the IEEE, Vol. 77, No. 4, 1989, pp. 541 – 580

[12] **Bogdan, S.**, **Lewis, F.L.**, **Kovačić, Z.**, **Mireles, J.**: *Manufacturing Systems Control Design – A Matrix-Based Approach*, Springer London, U.K., 2006, pp. 1290–1292.

[13] **Mireles, J.**, **Lewis, F.**: *Deadlock Analysis and Routing on Free-Choice Multipart Re-entrant Flow Lines Using a Matrix-Based Discrete Event Controller*, In Proceedings of the IEEE International Conference on Decision and Control, Vol. 1, 2002, pp. 793-798

[14] **Holland, J.H.**, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, 1975.

[15] **Cavalieri, S.**: *Petri Nets and Genetic Algorithms to Increase Productivity in FMS*, In Proceedings of 2nd Conf. on Knowledge-Based Intelligent Electronic Systems, Australia, 1998.

[16] **Chen, J.H.**, **Fu, L.C.**, **M.H. Lin**, **Huang, A.C.**: *Petri-Net and GA-Based Approach to Modelling, Scheduling, and Performance Evaluation for Wafer Fabrication*, IEEE

Transactions on Robotics and Automation, Vol. 17, No. 5, 2001, pp. 619-636

[17] **Bogdan, S.**, **Lewis, F.L.**: *Matrix Approach to Deadlock Avoidance of Dispatching in Multi-Class Finite Buffer Re-entrant Flow Lines, Intelligent Control*, in Proceedings of IEEE International Symposium, No. 16-18, 1997, pp. 397–402

[18] **Goncavles, J.F.**, **Mendes, J.M.**, **Resende, M.G.C.**: *A Hybrid Genetic Algorithm for the Job Shop Scheduling Problem*, European Journal of Operational Research, Vol. 167, 2005, pp. 77-95

[19] **Leon, V.J.**, et al.: *Strength and Adaptability of Problem-Space Based Neighbourhoods for Resource Constrained Scheduling*, Operations Research Spectrum, Vol. 17, 1995, pp. 173–182