

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
«ЗАПОРІЗЬКА ПОЛІТЕХНІКА»**

МЕТОДИЧНІ ВКАЗІВКИ

до виконання курсового проєкту з дисципліни
«Залізничний транспорт промислових підприємств»
для студентів спеціальності
275 «Транспортні технології (на залізничному транспорті)»
усіх форм навчання

Методичні вказівки до виконання курсового проекту з дисципліни «Залізничний транспорт промислових підприємств» для студентів спеціальності 275 «Транспортні технології (на залізничному транспорті)» усіх форм навчання /Укл.: С.М. Турпак, О.О. Острогляд, О.О. Падченко. - Запоріжжя: НУ «Запорізька політехніка», 2024. - 33 с.

Укладачі: С.М. Турпак, професор, д-р техн. наук;
О.О. Острогляд, канд. техн. наук;
О.О. Падченко, ст. викл.

Рецензент: О.Ф. Кузькін, професор, д-р техн. наук.

Відповідальний
за випуск: Т.В. Кальченко, зав. навч. лаб.

Затверджено на засіданні
кафедри «Транспортні технології»
протокол № 7
від 27 травня 2024 р.

Рекомендовано до видання
НМК транспортного факультету
протокол № 1
від 17 червня 2024 р.

ЗМІСТ

	с.
Вступ	4
1 Розробка концептуальної моделі системи перевезень рідкого шлаку.....	5
2 Розробка імітаційної моделі мікрологістичної системи перевезень рідкого шлаку.....	14
3 Аналіз результатів роботи моделі та вибір раціональних параметрів	28
4 Вибір оптимальних параметрів функціонування мікрологістичної системи перевезень шлаку.....	31
Перелік рекомендованої літератури	33

ВСТУП

Метою курсового проекту є допомогти студентам закріпити та поглибити знання з дисципліни, а також придбати навички у розв'язанні задач функціонування та взаємодії транспортних систем промислових підприємств на сучасному науково-технічному рівні, навчити їх здійснювати оптимальне управління та розробляти проектні рішення щодо удосконалення роботи транспортних систем промислових підприємств.

В ході роботи студенти проектують та організовують роботу транспортної системи металургійного підприємства.

Проект складається з розрахунково-пояснювальної записки обсягом 20...30 сторінок та графічної частини, яка може бути представлена у вигляді рисунку на форматі А1 або у вигляді слайдів.

Основна частина курсового проекту може складатися з однойменних розділів методичних вказівок до курсового проектування, починаючи з розділу 2, або близьких до них за змістом розділів з назвами, які, на думку автора, відображають їх змістове наповнення.

Здобувачам вищої освіти надається можливість вносити корективи до існуючої моделі або створювати свою, будь якого рівня деталізації.

В графічній частині «Аналіз результатів роботи моделі» рекомендовано розміщувати діаграми розподілу часу від постановки ковшів на фронт наливання рідкого шлаку до початку випуску, діаграми часу використання ресурсів «локомотиви» та «ковші», графіки залежності коефіцієнту використання локомотивів від кількості ковшів, результати економічного аналізу у табличному вигляді.

У висновках рекомендовано послідовно навести підсумки виконання кожного етапу курсового проекту.

1 РОЗРОБКА КОНЦЕПТУАЛЬНОЇ МОДЕЛІ СИСТЕМИ ПЕРЕВЕЗЕНЬ РІДКОГО ШЛАКУ

Технологічний процес перевезення рідкого чавуну та шлаку залізничним транспортом типовий на багатьох металургійних підприємствах. Складання концептуальної моделі починається з відображення матеріальних потоків рідкого чавуну та шлаку з печей. На рисунку 1.1 показана загальна концептуальна модель управління процесами випуску чавуну та шлаку доменними печами.



Рисунок 1.1 – Загальна концептуальна модель управління процесами випуску чавуну та шлаку доменними печами

Ця модель враховує вплив інших підсистем та зовнішніх факторів в системі залізничного транспорту підприємства.

В спрощеному вигляді модель виглядає наступним чином (рисунок 1.2).

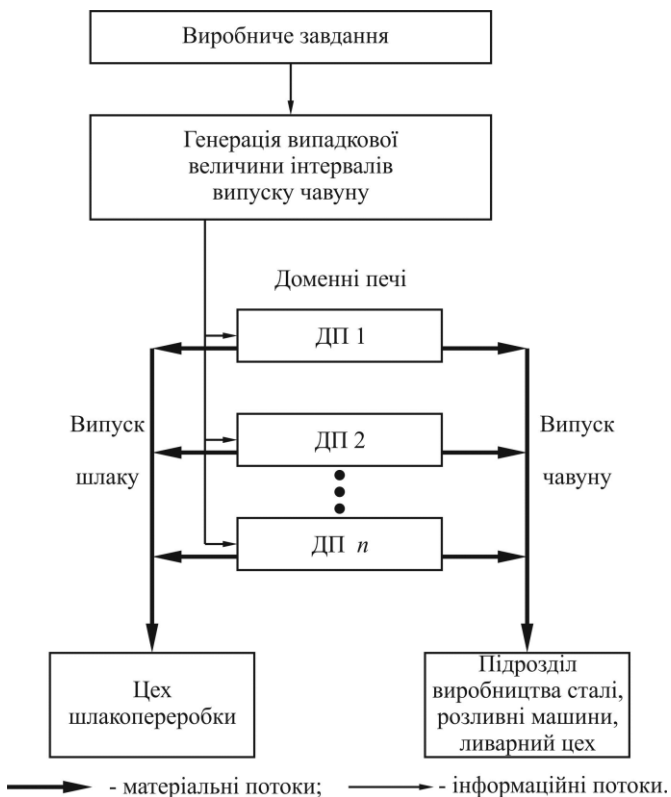


Рисунок 1.2 – Система компенсації впливу інших підсистем та зовнішніх факторів на процеси випуску чавуну та шлаку

Вплив інших підсистем та зовнішніх факторів компенсовано генерацією випусків чавуну за певним розподілом випадкової величини.

Імітаційна модель може бути складена зі стандартизованих елементів, які виконують певні операції по обслуговуванню замовлень в системі.

На рисунку 1.3 показана базова типова структура імітаційної моделі транспортної системи перевезень рідкого чавуну.

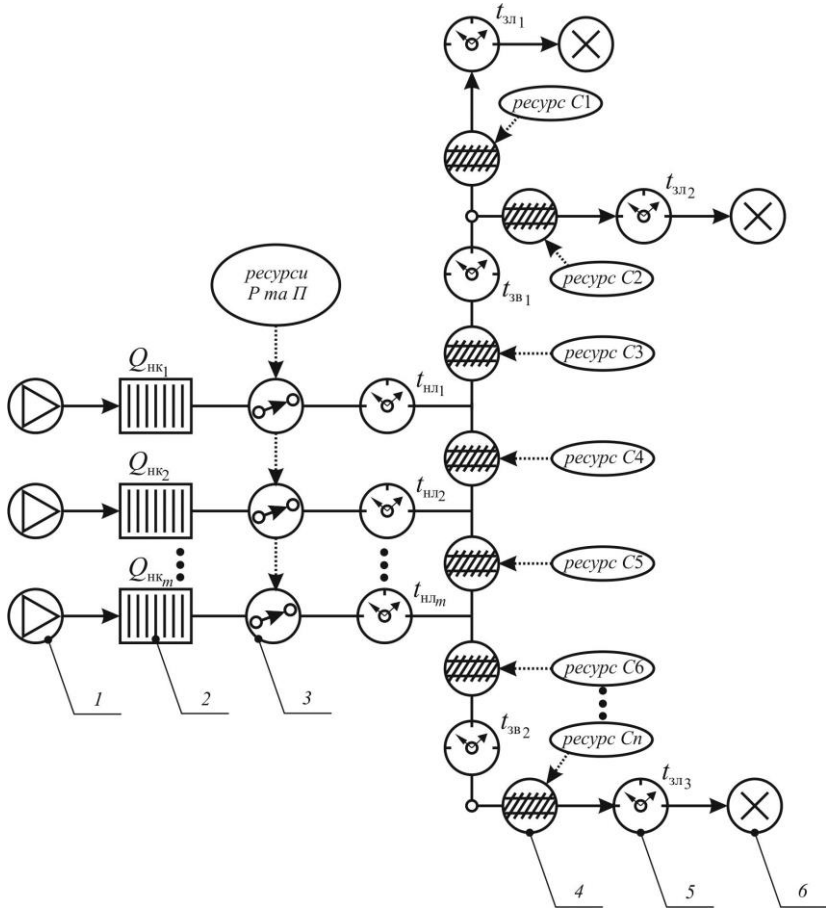


Рисунок 1.3 – Базова типова структура імітаційної моделі транспортної системи перевезень чавуну

1 – генератор замовлень; 2 – черга обслуговування; 3 – пристрій захвату рухомих та пересувних ресурсів; 4 – пристрій захвату стаціонарного ресурсу; 5 – пристрій обслуговування замовлень; 6 – знищувач замовлень.

За представленою структурою замовлення є вимогою на захват та переміщення ресурсів двох типів: Π – пересувних (ковші) та P – рухомих (локомотиви). Замовлення знаходяться у черзі до появи вільних ресурсів. Після їх захоплення (елемент 3), визначається напрям руху замовлень до пристроїв обслуговування – які імітують роботу вагових систем (обслуговування відбувається протягом часу $t_{зв1}, t_{зв2}$) та кінцевих споживачів, наприклад, ливарного цеху, міксерного відділення та розливних машин (обслуговування відбувається протягом часу $t_{зл1}, t_{зл2}, t_{зл3}$).

В процесі руху замовлення захоплює стаціонарні ресурси (типу С), які імітують окремі ділянки залізничних колій. Ділянки визначаються з огляду на заперечення одночасного знаходження на них більш одного замовлення.

Вважається, що у разі руху замовлень назустріч одне одному, у місцях стикування ділянок вони мають можливість пропуску одне одного (фактично, це відбувається шляхом заїзду по напрямку колій наливу чавуну та ін. окремих печей). Тобто, досягнувши межі ділянки (що може імітуватись створенням моделі транспортної мережі, або затриманням заявки на час, необхідний для проходження ділянки), замовлення звільняють відповідний захоплений ресурс, який може приєднатись до іншого, у тому числі, зустрічного замовлення.

Після знищення замовлень (чавун вважається переданим до відповідної ланки виробничого процесу), ресурси очікують на чергове захоплення новими замовленнями.

Даний алгоритм передбачає циклічне утворення замовлень за визначений період до початку їх першого обслуговування (наливу рідкого чавуну) протягом часу $t_{вл1}, t_{вл1}, \dots, t_{влm}$. Але можна створювати замовлення вже на момент вивільнення пересувних ресурсів (ковшів). Фактично, диспетчер може планувати використання порожніх ковшів, навіть, якщо потреба в них ще не встановлена.

У цьому випадку структура імітаційної моделі транспортної системи перевезень рідкого чавуну буде виглядати наступним чином (рисунок 1.4).

За такої структури моделі, нові замовлення генеруються по факту знищення попередньо обслугованих замовлень. Зрозуміло, що для запуску системи потрібне разове утворення декількох замовлень.

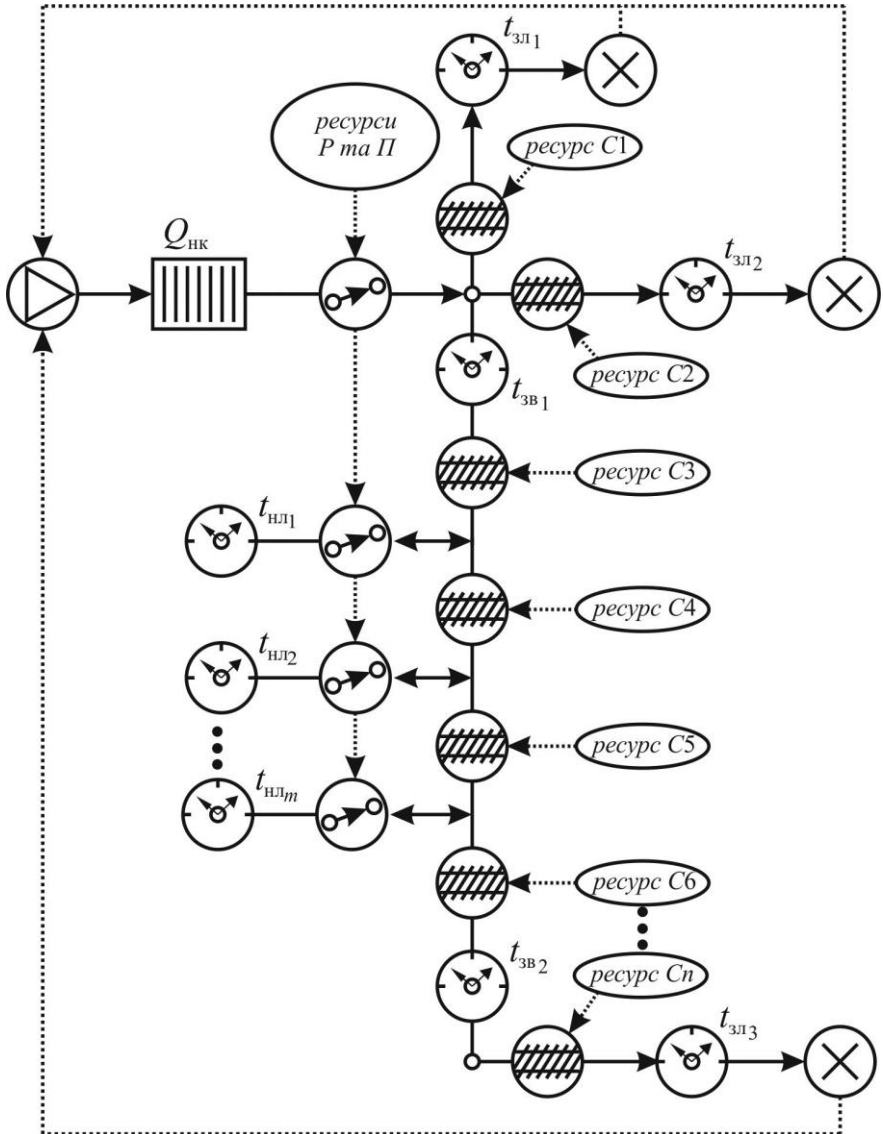


Рисунок 1.4 – Удосконалена структура імітаційної моделі транспортної системи перевезень рідкого чавуну

Для нових замовлень за відповідною логікою моделі обирається пристрій обслуговування (доменна піч). Вибір може здійснюватися за найменшим часом до початку випуску чавуну, або за найбільшою відповідністю кількості вивільнених ресурсів (при знищенні оброблених замовлень) потребі відповідної печі.

Замовлення захоплюють вільні рухомі та пересувні ресурси (ковші і локомотиви) та рухаються з ними, послідовно захоплюючи стаціонарні ресурси (ділянки колій). Досягнувши заданого пристрою обслуговування, замовлення звільняють рухомі ресурси (локомотиви) через час $t_{нл1}$, $t_{нл1}$, ... , $t_{нлm}$ (рисунок 1.4), а пересувні ресурси (ковші) утримують і після закінчення процесу обслуговування.

Після першого обслуговування, замовлення знову захоплює рухومی ресурс (локомотив) та рухається до наступних пристроїв, які імітують зважування (протягом часу $t_{зв1}$, $t_{зв2}$ – зазвичай, використовуються дві вагові) та зливання рідкого чавуну (за час $t_{зл1}$, $t_{зл2}$, $t_{зл3}$ – відповідно до продуктивності ливарного цеху, міксерного відділення або розливних машин).

При передаванні замовлень пристрою, звільняються рухомі ресурси. Тривалість вантажних операцій може встановлюватись шляхом аналітичних розрахунків або генерації за попередньо підібраним теоретичним законом розподілу випадкової величини.

Після цього обслуговування, замовлення знищують, а ресурси, які вони утримували (ковші), звільняють. По факту знищення утворюються нові заявки генератором замовлень і цикл робіт повторюється знову.

Базові моделі можна розвивати, досягаючи більш глибокого рівня деталізації та більш розвиненої логіки управління.

Так, при моделюванні роботи печі може враховуватись наявність похилої ванни, яка дозволяє виконувати пересування основної партії ковшів за рахунок тимчасового спрямування потоку рідкого металу до окремого ковша. Можна передбачати окремі цикли прибирання цього ковша за принципом врахування наповненості до вантажопідйомності та обмеженням часу початку твердіння чавуну.

Також можна враховувати використання резервних ковшів, які з деякою ймовірністю можуть заповнюватись чавуном.

Вплив інших підсистем на роботу транспортної системи, наприклад, перевезень чушкового чавуну, враховується необхідністю

відволікання локомотивів на ці перевезення. Зазвичай, «холодна» сторона розливних машин є окремим маневровим районом та її обслуговування не впливає на використання інших ресурсів в системі перевезень рідкого чавуну (ковшів та ділянок колій, по яких вони пересуваються).

Перевезення рідкого доменного шлаку відрізняються від перевезень рідкого чавуну утворенням составів з декількох плавок печей. Кожний состав перевозиться окремим локомотивом, який обслуговує їх на одному або декількох пунктах злиття.

Базова структура імітаційної моделі транспортної системи перевезень рідкого шлаку представлена на рисунку 1.5.

Відмінність даної моделі від показаної на рисунку 1.4 полягає у наявності елементів, позначених символами $Q_{\text{кв}}$ та $Q_{\text{ск}}$. Перший є чергою ковшів з вантажем, другий – імітує процес формування составів на відправлення.

Формування составів здійснюється за принципом досягнення у черзі $Q_{\text{кв}}$ такої кількості ковшів, яка відповідає, або перевищує розрахунковий розмір составу $Q_{\text{ск}}$ на відправлення (за тяговими розрахунками).

Крім того, состав формується з наявної кількості ковшів, якщо тривалість їх простою у завантаженому стані перевищує максимально припустиме значення $t_{\text{кв}} > t_{\text{max}}$.

Час обороту составу може бути представлений детально по операціях:

- рух до басейну грануляції;
- маневрова робота з постановки ковшів на фронт злиття шлаку;
- зливання ковшів у басейн;
- формування составу;
- рух составу до відвалу;
- маневрова робота з постановки ковшів на фронт вибивання;
- зливання залишків та вибивання ковшів;
- формування составу з порожніх ковшів;
- рух составу на станцію навантаження;
- обприскування внутрішньої поверхні ковшів;
- приймальні операції.

Можна об'єднати всі вищевказані операції та задати фіксовану тривалість обороту составу або шляхом генерації за теоретичним законом розподілу випадкової величини.

Після закінчення обробки замовлень вони знищуються та здійснюється визивання нових замовлень.

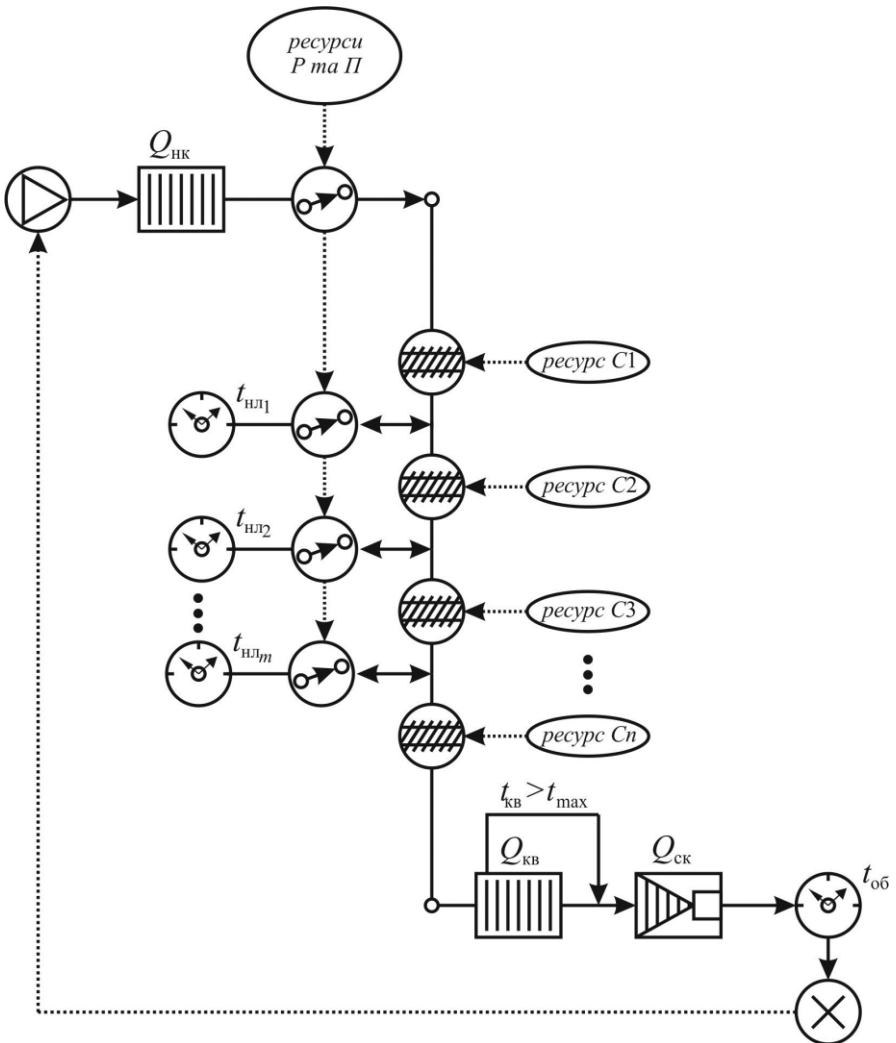


Рисунок 1.5 – Базова структура імітаційної моделі транспортної системи перевезень рідкого шлаку

Таким чином, в моделях транспортної системи перевезень продукції першого переділу металургійного підприємства в якості замовлень виступають вимоги на подавання порожніх ковшів до печей. Кількість необхідних пересувних ресурсів (ковшів) для обслуговування окремого замовлення, встановлюється за фіксованим або змодельованим значенням випадкової величини.

Для руху замовлення разом з захопленими ним пересувними ресурсами використовується рухомий ресурс (локомотив).

Просування по ділянках колій, де можливі перетинання, тобто конфлікти, при русі замовлень, здійснюється шляхом захоплення стаціонарних ресурсів.

Практична реалізація моделі може бути виконана за допомогою програм імітаційного моделювання. Такі програми дозволяють створювати імітаційні моделі за допомогою стандартних елементів та спостерігати за ходом моделювання у вигляді анімації процесів та змінювати параметри роботи в діалоговому режимі.

Згідно матеріалів, викладених у даному розділі, у відповідності до завдання на курсовий проект, необхідно розробити (із детальними поясненнями):

- концептуальну модель транспортної системи (за основу приймається модель, зображена на рисунку 1.2);
- структуру імітаційної моделі транспортної системи (за основу приймається модель, зображена на рисунку 1.5).

Крім того, в цьому розділі виконується статистичний аналіз вихідних даних для проектування та встановлюються теоретичні закони розподілу випадкової величини інтервалів між послідовними випусками шлаку доменними печами та параметри, які визначають форму та масштаб цих розподілів [1,2].

Отримані параметри використовуються надалі в імітаційній моделі мікрологістичної системи (МЛС) перевезень рідкого шлаку.

2 РОЗРОБКА ІМІТАЦІЙНОЇ МОДЕЛІ МІКРОЛОГІСТИЧНОЇ СИСТЕМИ ПЕРЕВЕЗЕНЬ РІДКОГО ШЛАКУ

В цьому розділі пояснюється процес створення моделі, в яку потрібно буде внести певні корегування в залежності від завдання. Він починається з формування схеми колійного розвитку станції (рисунок 2.1).

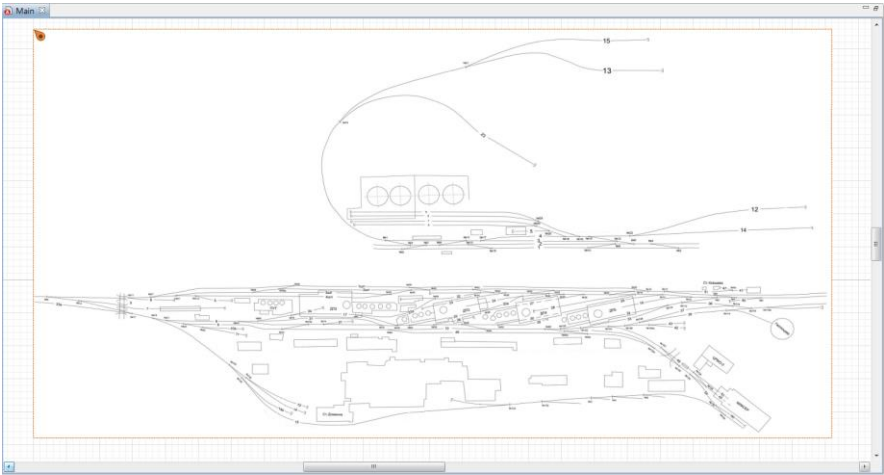


Рисунок 2.1 – Схема колійного розвитку

Створюється анімація моделі, так як саме вона визначатиме структуру моделі – її транспортну мережу.

Топологія транспортної мережі задається групою фігур: прямокутники задають вузли мережі, а лінії – зв'язки між ними, що відіграють роль шляхів руху заявок та ресурсів у модельованому просторі.

Для створення анімації використовується презентаційна палітра, яка містить набір елементарних фігур, за допомогою яких можна створити відповідне зображення.

Поверх схеми колійного розвитку прокладається транспортна мережа, по якій будуть рухатися заявки та ресурси та створюється

топология транспортной сети модели МЛС перевозень рідкого доменного шлаку (рисунок 2.2).



Рисунок 2.2 – Топология транспортной сети модели МЛС перевозень рідкого доменного шлаку

Для створення транспортної мережі використовуємо об'єкти типу «лінія» та «прямокутник». За допомогою «ліній» відображаємо перегони, станційні колії, «прямокутниками» відображаємо стрілочні перекладки, вантажні пункти, базові місця знаходження ресурсів – локомотивів, ковшів.

Додаємо на діаграму об'єкт типу *Network* (рисунок 2.3). Цей об'єкт задає властивості транспортної мережі моделі (змінимо стандартну назву *Network* на *Network_shlak*).

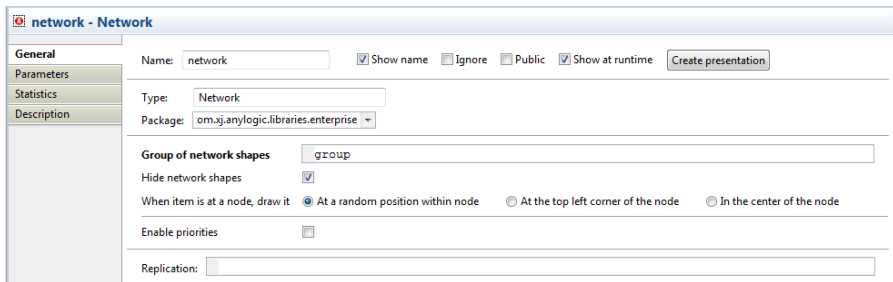


Рисунок 2.3 – Створення об'єкта *Network_shlak*

Тепер додаємо об'єкти, що задають мережеві ресурси. Мережеві ресурси можуть бути трьох видів: рухомі, переносні і статичні. У нашому випадку локомотиви будуть задані рухомими ресурсами, а ковші переносними. Окремі ділянки колій будуть відповідати статичним ресурсам.

Додаємо на діаграму класу *Main* об'єкти бібліотеки *Enterprise Library* типу *NetworkResourcePool*.

Об'єкти *ResourceLocShlak* та *ResourceKovshShlak* типу *NetworkResourcePool* описують, відповідно, рухомі (локомотиви) та переносні (ковші) ресурси (рисунок 2.4). Інші ресурси, задіяні в мережі, які показані на рисунку 2.4, статичні та імітують зайняття окремих ділянок колій.

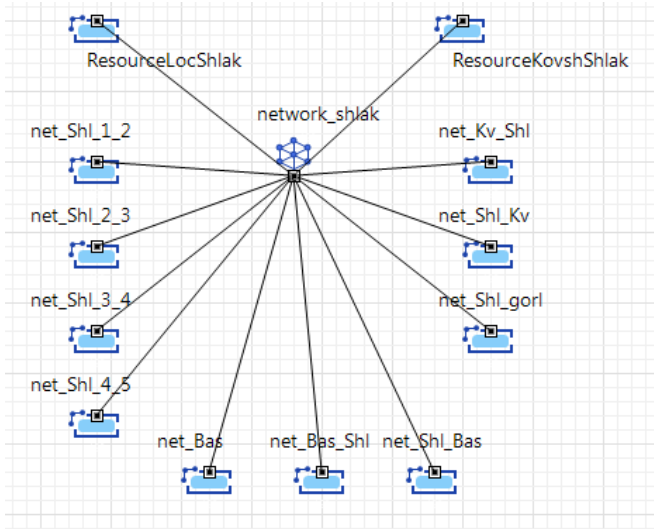


Рисунок 2.4 – Ресурси транспортної мережі імітаційної моделі

В формі, показаній на рисунку 2.5, для кожної групи ресурсів, приєднаних до мережі *Network_shlak* вказується тип, кількість ресурсів даного типу, швидкість їх переміщення (в умовних одиницях), місце базового розташування на мережі, фігура анімації та ін.

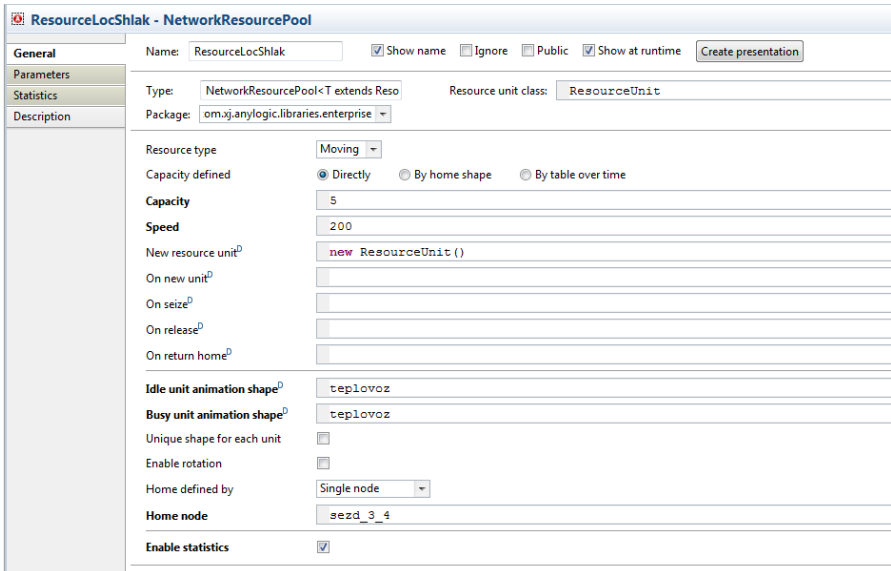


Рисунок 2.5 – Ресурси транспортної мережі імітаційної моделі

Своєрідним «серцем» моделі є блок генерації подій (рисунок 2.6).

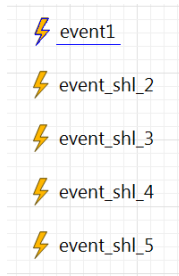
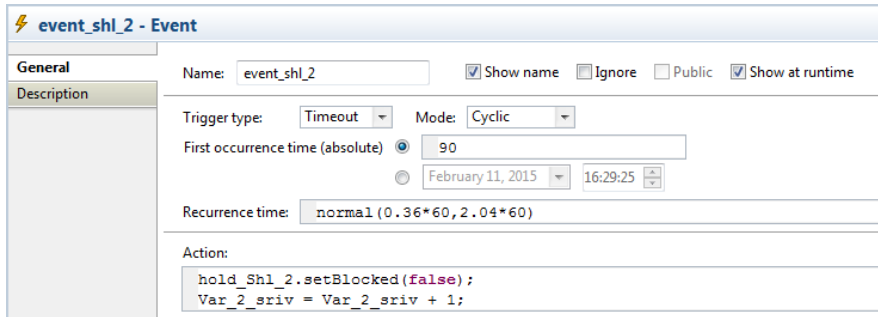


Рисунок 2.6 – Блок генерації подій моделі

Кожна подія *event_shl_2* - *event_shl_5* імітує закінчення випуску шлаку відповідною піччю №2 - №5.

Час подій, та їх дії задаються формою, представленою на рисунку 2.7.



event_shl_2 - Event

General

Name: Show name Ignore Public Show at runtime

Trigger type: Mode:

First occurrence time (absolute) 90 February 11, 2015 16:29:25

Recurrence time:

Action:

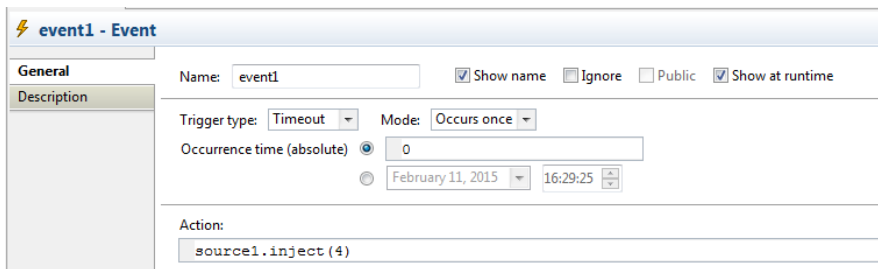
```
hold_shl_2.setBlocked(false);
Var_2_sriv = Var_2_sriv + 1;
```

Рисунок 2.7 – Опис події в імітаційній моделі

Так, в даній формі зазначено, що подія відбувається через визначений час (за «таймаутом») циклічно. Період циклу заданий функцією нормального розподілу випадкової величини за параметрами, визначеними для кожної печі в розділі статистичного аналізу дипломного проекту.

При настанні певної події виконуються наступні дії: розблокування елемента затримки руху замовлень у відповідній гілці моделі та перевірка своєчасного знаходження ресурсів (ковшів) на початок випуску шлаку. У разі зриву випуску плавки за відсутності ковшів, цей факт фіксується збільшенням змінних `Var_2_sriv`, `Var_3_sriv`, `Var_4_sriv`, `Var_5_sriv` – по кожній печі.

Подія `event_1` необхідна лише для початку роботи програми (виконується один раз за весь час роботи програми). Вона викликає чотири перших замовлення на подавання порожніх ковшів до печей (рисунок 2.8).



event1 - Event

General

Name: Show name Ignore Public Show at runtime

Trigger type: Mode:

Occurrence time (absolute) 0 February 11, 2015 16:29:25

Action:

```
source1.inject(4)
```

Рисунок 2.8 – Опис події виклику перших замовлень в моделі

Кожне замовлення викликає та захоплює до себе вільний рухомий ресурс – локомотив, потім разом з ним захоплюються також переносні ресурси – ковші. Замовлення рухаються по певних гілках програми (їх структура буде розглянута пізніше) разом з захопленими ресурсами – «локомотивом» та «ковшами» до елемента, який позначає місцезнаходження доменної печі. Ресурс «локомотив» звільняється – від'єднується від замовлення, а ресурси «ковші» - залишаються разом з замовленням та очікують подію «закінчення плавки».

Після настання цієї події, замовлення з ресурсами «ковші» знову захоплює ресурс «локомотив» для руху в місце формування составу на відправлення зі станції Ковшова, і в цей момент в моделі генеруються нові замовлення на обслуговування доменної печі.

Таким чином, технологічний процес видачі шлаку з доменних печей виконується в моделі незалежно від транспортного обслуговування – за графіком випусків з урахуванням випадкового характеру відхилень від нього. Транспортне обслуговування орієнтоване на виконання замовлень щодо подачі шлаковозів під печі, які утворюються в системі відразу по закінченню плавки.

На рисунку 2.9 представлена основна частина моделі, яка імітує логіку проходження та обробки замовлень.

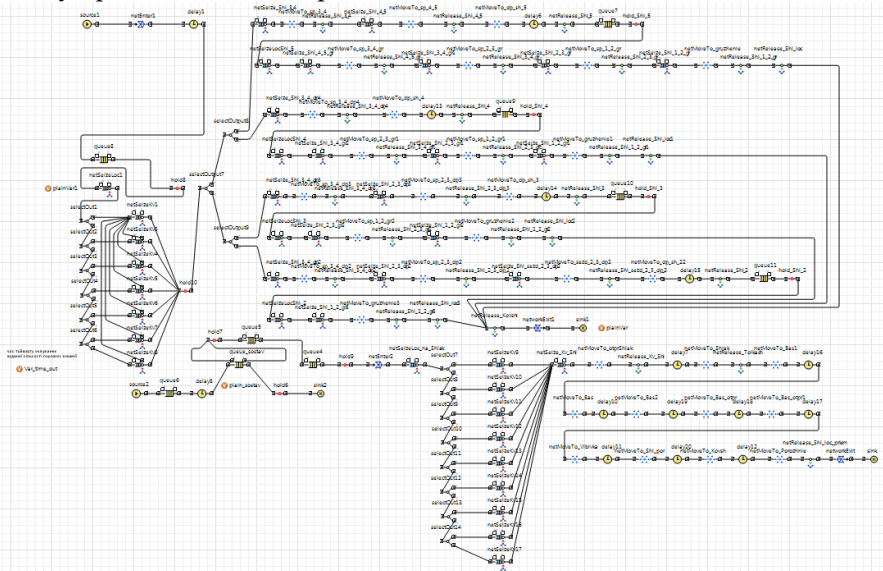


Рисунок 2.9 – Основна частина моделі

Модель складена з таких елементів.
Об'єкт *Source* (рисунок 2.10).

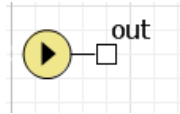


Рисунок 2.10 – Елемент моделі *Source*

Створює заявки. Зазвичай використовується в якості початкової точки потоку заявок. Заявки можуть створюватися згідно заданої інтенсивності (яка може змінюватися динамічно за допомогою методу *set_rate*, часу між прибуттям, що змінюється в часі інтенсивності, заданої за допомогою табличної функції (таблиці інтенсивності), розкладом, що задає точний час і кількість заявок, або шляхом виклику методу об'єкта *inject*, як це здійснюється в нашому випадку при настанні циклічних подій.

Об'єкт *Delay* (рисунок 2.11).

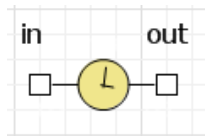
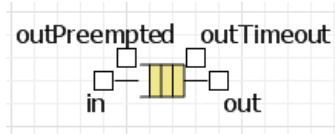


Рисунок 2.11 – Елемент моделі *Delay*

Затримує заявки на заданий період часу. Час затримки обчислюється динамічно, може бути випадковим, залежати від поточної заявки або від якихось інших умов. Одночасно можуть бути затримані відразу кілька заявок (не більше заданої місткості об'єкта *capacity*). Заявки затримуються незалежно один від одного - час затримки обчислюється окремо для кожної заявки. Як тільки час затримки закінчується, заявка залишає об'єкт. Якщо об'єкт *Delay* заповнений повністю, то нову заявку він не прийме, і в цьому випадку потрібно буде помістити перед ним спеціальний об'єкт буферизації, наприклад, *Queue*.

Об'єкт *Queue* (рисунок 2.12).

Рисунок 2.12 – Елемент моделі *Queue*

Об'єкт *Queue* моделює чергу заявок, які очікують прийому об'єктами, наступними за даними, або ж є сховищем заявок загального призначення. Можна задати максимальний час очікування заявки в черзі.

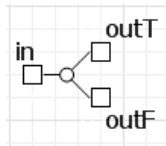
Заявка може покинути об'єкт *Queue* різними способами:

- через порт *out*, коли об'єкт, який є наступним в блок-схемі за об'єктом *Queue*, готовий прийняти заявку;
- через порт *outTimeout*, якщо заявка проведе в черзі задану кількість часу (якщо включений режим таймаута);
- через порт *outPreempted*, під тиском наступної заявки при заповненій черзі (якщо включений режим витискання);
- «вручну», шляхом виклику функції *remove* або *removeFirst*.

У першому випадку об'єкт *Queue* покидає заявка, знаходиться в самому початку черги (в нульовій позиції). Якщо заявка спрямована в порт *outTimeout* або *outPreempted*, то вона повинна покинути об'єкт миттєво.

Якщо включена опція витіснення, то об'єкт *Queue* завжди готовий прийняти нову заявку, в іншому випадку при заповненій черзі заявка прийнята не буде.

Об'єкт *SelectOutput* (рисунок 2.13).

Рисунок 2.13 – Елемент моделі *SelectOutput*

Об'єкт направляє вхідні заявки в один із двох вихідних портів залежно від виконання заданого (детерміністичного або заданого за допомогою ймовірностей) умови.

Умова може залежати як від заявки, так і від якихось зовнішніх факторів. Заявка, що надійшла, залишає об'єкт у той же момент часу. Може використовуватися для сортування заявок відповідно до заданого критерію, для випадкового поділу потоку заявок на частині й т. ін.

Об'єкт *NetworkEnter* (рисунок 2.14).

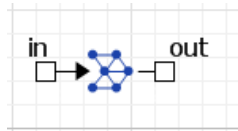


Рисунок 2.14 – Елемент моделі *NetworkEnter*

Використовується при моделюванні транспортних мереж. Реєструє заявку в мережі і поміщає її в заданий вузол мережі. Після додавання в мережу заявка може переміщатися по мережі і використовувати мережеві ресурси. Операція додавання в мережу займає нульовий час.

Об'єкт *Hold* (рисунок 2.15).

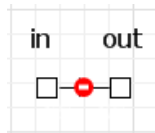


Рисунок 2.15 – Елемент моделі *Hold*

Цей елемент блокує або розблоковує потік заявок на певній ділянці блок-схеми. Якщо об'єкт знаходиться в заблокованому стані, то заявки не надходять на його вхідний порт і чекатимуть, поки об'єкт не буде розблокований. Станом об'єкта можна управляти програмно за допомогою методу *setBlocked*. В розробленій моделі дані об'єкти блокують замовлення з ресурсами типу «ковші», які

знаходяться в черзі (імітація наливу шлаку) до настання події закінчення випуску шлаку.

Об'єкт *NetworkSeize* (рисунок 2.16).

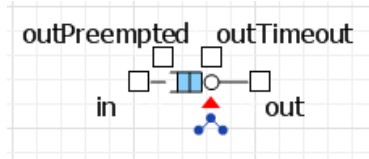


Рисунок 2.16 – Елемент моделі *NetworkSeize*

Використовується при моделюванні транспортних мереж. Об'єкт захоплює для заявки задану кількість мережевих ресурсів. При необхідності може пересилати захоплені ресурси в заданий місце мережі та приєднувати їх до заявки. Об'єкт можна розглядати як чергу *Queue* заявок, які очікують ресурсів, за якою можуть слідувати об'єкти *NetworkSendTo* і *NetworkAttach*. Спочатку ресурси запитуються для першої заявки з черги, і поки ця заявка не захопить ресурси, ресурси для наступних заявок не виділяються.

Об'єкт *NetworkMoveTo* (рисунок 2.17).

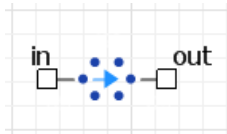


Рисунок 2.17 – Елемент моделі *NetworkMoveTo*

Використовується при моделюванні транспортних мереж. Переміщує заявку в нове місце мережі. Якщо до заявки приєднані якісь ресурси, то вони переміщуються разом із заявкою. При цьому незалежно від швидкості ресурсів переміщатися така група із заявки та ресурсів буде зі швидкістю заявки. Час, який заявка проведе в цьому об'єкті, дорівнюватиме довжині найкоротшого з можливих шляхів з поточного місця розташування заявки у вузол призначення, поділеної на швидкість заявки. Заявка буде відображатися на анімації мережі, що рухається уздовж обраного маршруту.

Об'єкт *NetworkRelease* (рисунок 2.18).

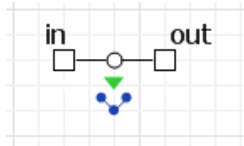


Рисунок 2.18 – Елемент моделі *NetworkRelease*

Використовується при моделюванні транспортних мереж. Звільняє всі або якісь певні мережеві ресурси (раніше захоплені заявкою за допомогою об'єкту *NetworkSeize*). У разі звільнення якихось певних ресурсів, вони вибираються із загального числа захоплених ресурсів згідно заданому списку. Вся операція займає нульовий час.

Об'єкт *NetworkExit* (рисунок 2.19).

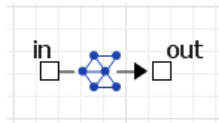


Рисунок 2.19 – Елемент моделі *NetworkExit*

Використовується при моделюванні транспортних мереж. Видаляє заявку з мережі. Заявка при цьому перестає відображатися на анімації мережі.

Об'єкт *Sink* (рисунок 2.20). Знищує заявки. Зазвичай використовується в якості кінцевої точки потоку заявок.

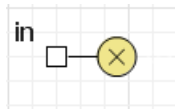


Рисунок 2.20 – Елемент моделі *Sink*

Для зручності подальшого опису програми розділимо основну частину моделі на три блоки (рисунок 2.21).

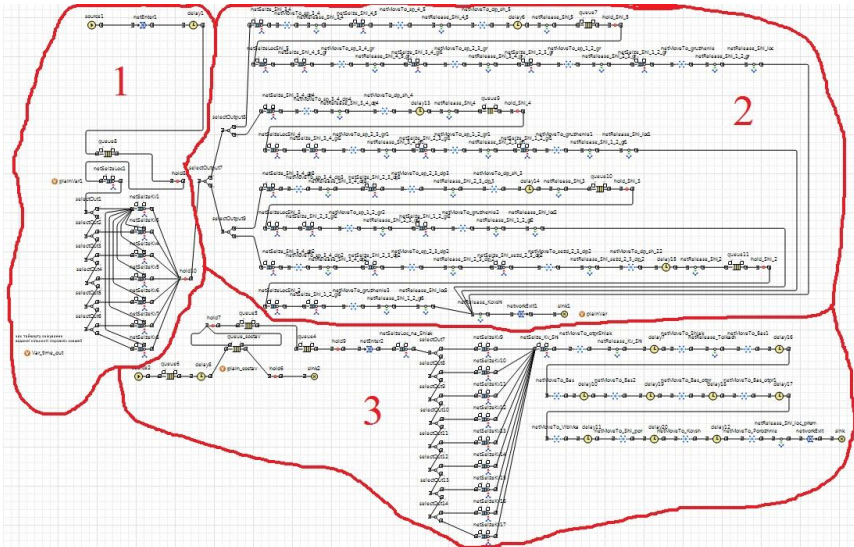


Рисунок 2.21 – Блоки основної частини моделі

В першому блоці утворюються замовлення, які захоплюють вільний ресурс «локомотив» (у разі його відсутності, очікують на нього в черзі). Потім за законом розподілу випадкової величини замовленої кількості ковшів за допомогою елементів *SelectOutput* замовлення направляється через відповідний елемент типу *NetworkSeize*, де вказано, скільки ресурсів необхідно захватити шляхом перерахування відповідної кількості їх назв (рисунок 2.22 – поле «список ресурсів»).

В нашому випадку передбачена можливість виходу замовлення з об'єкту *NetworkSeize* не тільки при захопленні потрібної кількості ресурсу «ковші», а і у випадку відсутності протягом тривалого часу потрібної кількості цих ресурсів (вихід «за таймаутом»). В цьому випадку захоплюється хоча б один ківш та утворюється додаткове замовлення на надолуження їх потрібної кількості.

Надалі замовлення направляється до тієї печі, час випуску шлаку в якій наступить скоріше, ніж в інших.

netSeizeKv3 - NetworkSeize

General

Name: netSeizeKv3 Show name Ignore Public Show at runtime

Type: NetworkSeize<T extends Entity> Entity class: Entity

Package: om.xj.anylogic.libraries.enterprise

List of resources { pool1, ... }^D {ResourceKovshShlak, ResourceKovshShlak}

On enter^D

On exit^D

Maximum queue capacity

Enable exit on timeout

Timeout^D Var_time_out

On exit (timeout)^D source1.inject(1)

Enable preemption

Send seized resources

Attach seized resources

Animation guide shape (queue)

Animation type (queue) Path

Animation direction (queue) Forward Backward

Enable statistics

Рисунок 2.22 – Параметри об'єкта *NetworkSeize*

Таким чином, до другого блоку програми замовлення надходить до відповідної гілки обслуговування окремої печі. Ці гілки відрізняються шляхом прямування замовлень. По мірі просування до печі, замовлення разом з захопленими ресурсами («локомотивом» і «ковшами») захоплюють статичні ресурси – ділянки колій, та рухаються по них.

На окремі ділянки не може знаходитись більше одного составу, тобто одна ділянка – це один окремий ресурс. Коли замовлення переходить на нову ділянку, попередньо захоплена ділянка звільняється та може бути захоплена іншим замовленням. У разі зустрічного руху замовлень вони наближаються до межі захоплених ділянок, і коли зустрінуться у цьому місці, зможуть, по-суті, обміняти своїми ділянками (звільнити «свою» та захопити «чужу»).

Стосовно реальних об'єктів, ця ситуація розуміється наступним чином: місцем стикування ділянок є стрілочні переводи, які з'єднують три ділянки, тому один з составів може переїхати на вільну ділянку, пропустити зустрічний состав по стрілці та зайняти колію, яку він вивільнив.

По надходженні навантажених ресурсів-«ковшів» до місця накопичення, вони звільняються від замовлення. Це замовлення знищується об'єктом *sink1* та відразу цей об'єкт викликає утворення об'єктом *source2* нової групи замовлень за кількістю ковшів, які рухались зі знищеним замовленням. Ці дії обумовлені технічними особливостями програми і необхідні для реалізації третього блоку, який забезпечує імітацію руху составів, які обмежені по кількості ковшів та іноді потребують розділення навантажених окремими печами груп.

В третьому блоці програми визначається розмір составу: або при досягненні максимального розміру в черзі накопичення *queue_sostav*, або при перевищенні припустимого терміну знаходження «ковшів» в цій черзі (вихід «за таймаутом»). В цьому блоці одне замовлення відповідає одному «ковшу». Після формування составу, фіксується його розмір; замовлення, знищуються об'єктом *sink2*, окрім одного, яке надалі захоплює відповідну розміру составу кількість ресурсів. Тобто це залишене замовлення імітує рух одного составу з декількома ресурсами – «ковшами».

Замовлення рухається по мережі станція Ковшева – станція Шлакова – станція Гранбасейн – станція Шлакова – станція Ковшева, потім звільняє всі захоплені нею ресурси та знищується об'єктом *sink*.

Для реалізації даного розділу необхідно:

- встановити java – файл «jre-7u21-windows-x64» (якщо була встановлена інша версія – бажано її спочатку видалити). Це дозволяє уникнути блокування роботі аплету Anylogic через можливість зниження рівня безпеки;

- розкрити архів «аплет 07_11.rar» та запустити в папці «operausb1218» файл «opera.exe»;

- у програмі «opera» натиснути комбінацію «ctrl + O», обрати файл (аплет) «Kovshevaya_parameter_07_11_2022.alp».

- ввести параметри моделювання та виконати прогони моделі (визначення закону розподілу послідовних інтервалів випуску шлаку здійснюється шляхом статистичного аналізу вибірок даних згідно варіанту завдання).

Необхідні файли надаються викладачем через систему «Moodle: НУЗП» (<https://moodle.zp.edu.ua/>) або в інший зручний спосіб.

3 АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ МОДЕЛІ ТА ВИБІР РАЦІОНАЛЬНИХ ПАРАМЕТРІВ

Головними результатами роботи моделі є такі показники:

- час використання ресурсу «локомотиви»;
- час використання ресурсу «ковші»;
- кількість зафіксованих зривів в роботі печей з причини несвоєчасної постановки ковшів.

Додатково можна аналізувати:

- розподіл часу від постановки ковшів на фронт наливання рідкого шлаку до початку випуску;
- час обороту составів та ін.

Статистичні дані збираються спеціально розробленими об'єктами програми та відображаються у вигляді діаграм (рисунок 3.1 та 3.2).

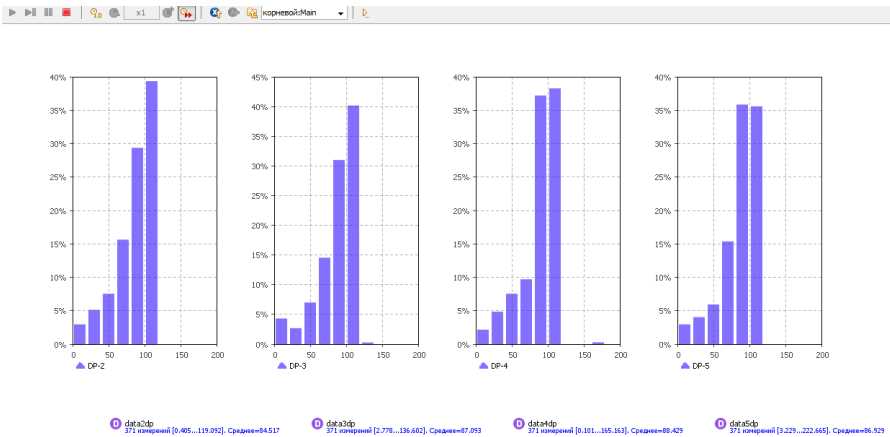


Рисунок 3.1 – Діаграми розподілу часу від постановки ковшів на фронт наливання рідкого шлаку до початку випуску

Імітація роботи здійснюється за хвилинним масштабом часу протягом місяця роботи.



Рисунок 3.2 – Діаграми часу використання ресурсів «локомотиви» та «ковші»

Перед кожними прогонами змінюються параметри кількості ресурсів: ковшів та локомотивів (кількість прогонів, шаг зміни параметрів та їх діапазони узгоджуються з викладачем на консультаціях в лабораторії кафедри транспортних технологій).

Результати роботи моделі наводяться у вигляді таблиці 3.1.

Таблиця 3.1 – Результати роботи моделі

Номер експерименту	Параметри		Результати		
	Кількість локомотивів	Кількість ковшів	Ступінь використання локомотивів	Ступінь використання ковшів	Зриви в обслуговуванні цеху, %

Результати роботи повинні бути представлені також у графічному вигляді.

На рисунку 3.3 наведений приклад аналізу результатів визначення параметрів МЛС перевезень рідкого шлаку (затемнені області характеризують зростання ймовірності зривів в обслуговуванні цехів).

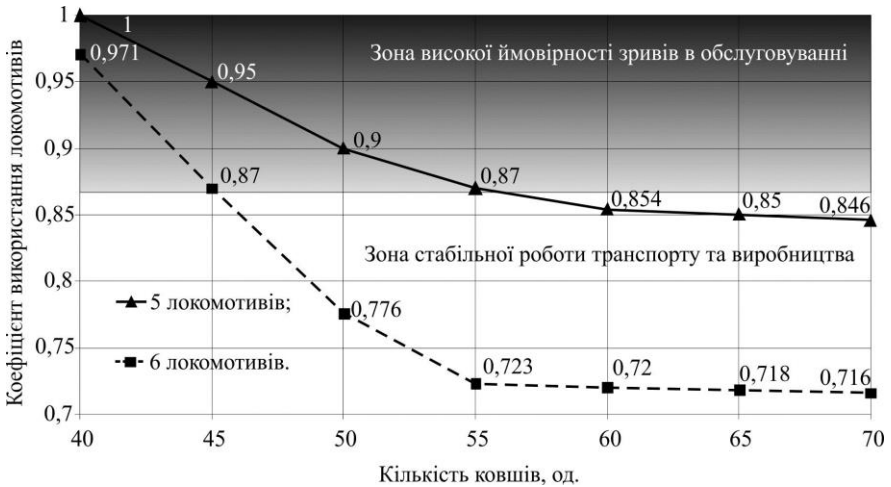


Рисунок 3.3 – Графіки залежності коефіцієнту використання локомотивів від кількості ковшів

З аналізу результатів графіків на рисунку 3.3 можна зробити наступні висновки:

- при використанні в роботі 4 локомотивів кількість зривів в обслуговуванні доменного цеху дуже високе. Збільшення кількості ковшів не призводить до суттєвого покращення ситуації;
- при використанні в роботі 5 локомотивів необхідно не менше 56 ковшів, щоб уникнути зривів в роботі;
- при використанні в роботі 6 локомотивів достатньо 45 ковшів.

В даному розділі наводяться діаграми розподілу часу від постановки ковшів на фронт наливання рідкого шлаку до початку випуску, діаграми часу використання ресурсів «локомотиви» та «ковші», графіки залежності коефіцієнту використання локомотивів від кількості ковшів.

З аналізу графіків формуються можливі варіанти організації перевезень, по кожному з них необхідно виконати економічні розрахунки для визначення параметрів функціонування транспортної системи, які забезпечують її найбільшу ефективність.

4 ВИБІР ОПТИМАЛЬНИХ ПАРАМЕТРІВ ФУНКЦІОНУВАННЯ МІКРОЛОГІСТИЧНОЇ СИСТЕМИ ПЕРЕВЕЗЕНЬ ШЛАКУ

Експлуатаційні витрати по варіантах організації робіт визначаються за формулою:

$$C_{\text{заг}} = C_3 + C_c + C_n + C_{\text{лок}} + C_{\text{кв}}, \quad (4.1)$$

де C_3 - основна і додаткова зарплата працівників, грн;
 C_c - відрахування на соціальне страхування, грн;
 C_n - накладні витрати, грн;
 $C_{\text{лок}}$ - витрати на роботу локомотивів, грн;
 $C_{\text{кв}}$ - витрати на утримання шлаковозних ковшів, грн.

Заробітну плату машиніста тепловоза не враховуємо, оскільки вона входить до собівартості локомотиво-години маневрової роботи локомотива (тепловоз іноді може працювати без помічника або без складача поїздів).

Нарахування заробітної плати помічника машиніста тепловоза здійснюється за формулою:

$$C_3 = N \cdot k_{\text{пр}} \cdot C_{\text{год}} [\Phi_p (k_n + k_{\text{шк}}) + \Phi_{\text{св}}], \quad (4.2)$$

де N - кількість працівників (4 на кожен локомотивів в роботі);
 $k_{\text{пр}}$ - коефіцієнт, який враховує премію, $k_{\text{пр}} = 1,3$;
 $C_{\text{год}}$ - годинна тарифна ставка, $C_{\text{год}} = 0,5$ у.о./год.;
 Φ_p - фонд робочого часу за рік, $\Phi_p = 2002$ год.;
 k_n - коефіцієнт, який враховує доплату за роботу у вечірній та нічний час, $k_n = 1,2$;
 $k_{\text{шк}}$ - коефіцієнт, який враховує доплату за роботу у шкідливих умовах, $k_{\text{шк}} = 0,1$.
 $\Phi_{\text{св}}$ - час роботи протягом року у святкові дні, $\Phi_{\text{св}} = 60$ год.

Відрахування на соціальне страхування приймаємо у розмірі 22% від фонду заробітної плати.

Накладні витрати приймаються у розмірі 34% від фонду оплати

праці.

Витрати на роботу локомотивів визначаються за формулою:

$$C_{\text{лок}} = N_{\text{лок}} \cdot T_{\text{лок}} \cdot C_{\text{лг}}, \quad (4.3)$$

де $N_{\text{лок}}$ - кількість локомотивів, од.;

$T_{\text{лок}}$ - час роботи локомотива протягом року.

$C_{\text{лг}}$ - вартість локомотиво-години, $C_{\text{лг}} = 15$ у.о./год.

Витрати на утримання шлаковозних ковшів визначаються за формулою:

$$C_{\text{кв}} = 365 \cdot N_{\text{кв}} \cdot C_{\text{кв}}, \quad (4.4)$$

де $N_{\text{кв}}$ - кількість шлаковозних ковшів робочого парку, од.;

$C_{\text{кв}}$ - вартість годино-добу утримання одного ковша, $C_{\text{кв}} = 30$ у.о./добу.

Виконані розрахунки $C_{\text{заг}}$ порівнюються по варіантах організації робіт та з них обирається найбільш економічний.

Наводяться висновки щодо оптимальних параметрів функціонування транспортної системи.

ПЕРЕЛІК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Лашених, О.А. Методи і моделі оптимізації транспортних процесів і систем / О.А. Лашених, О.Ф. Кузькін. – Запоріжжя: ЗНТУ, 2006. – 435 с.
2. Кузькін, О.Ф. Імовірнісні і статистико-експериментальні методи аналізу транспортних систем : навчальний посібник / О.Ф. Кузькін, О.А. Лашених, С.М. Турпак. – Запоріжжя: ЗНТУ, 2013. – 371с.
3. Турпак, С.М. Методи та моделі управління вагонопотоками на металургійних підприємствах : монографія / С. М. Турпак. – Херсон : Грінь Д.С., 2014. – 146 с.
4. Турпак, С.М. Логістичні системи управління залізничним транспортом металургійних підприємств : монографія / С. М. Турпак. – Херсон : Грінь Д. С., 2015. – 264 с.
5. The AnyLogic Company. URL: <https://www.anylogic.com/company/about-us/>.