

**П р о г р а м н е
з а б е з п е ч е н н я Arena
і м і т а ц і й н о г о
м о д е л ю в а н н я с и с т е м**

- Arena Simulation Software— розробка компанії Systems Modeling Arena
На сьогодні цим софтом займається компанія Rockwell Automation
<https://www.rockwellautomation.com/en-us/products/software/arena-simulation/buying-options/download.html>
- Основа технологій Arena - мова моделювання SIMAN і система Cinema Animation. SIMAN (перша реалізація 1982р.)
- Переваги Arena: 1) модель будується з конструктивних блоків, набір яких достатньо широкий, 2) ієрархічна модель будується дуже просто і наочно, 3) анімаційні засоби представлені широко (від примітивних під час запуску моделі до графічних з анімацією персонажів і елементів моделі), 4) широкий спектр звітів з моделювання, 5) набір інструментів для статистичного збору інформації та її графічного представлення.
- Аналогічний підхід використовує програмне забезпечення Simio (з більш сучасною анімацією та більшою швидкістю)

Відеоресурси Arena

- Тьюторіал для початківців
<https://www.youtube.com/watch?v=dlbW8WFen1s>
- Відео з демонстрацією анімаційних можливостей Arena
<https://www.youtube.com/watch?v=6EFPB0FUgCM>
- Відео з демонстрацією анімаційних можливостей Simio
<https://www.simio.com/applications/aerospace-and-defense-simulation-and-scheduling-software/index.php>

Інтерфейс системи Arena

Вікно додатку розділене на три області:

- вікно робочого поля;
- вікно властивостей модулів;
- вікно проекту.

Вікно проекту включає декілька панелей:

- Basic Process (панель основних процесів) - містить модулі, які використовуються для моделювання;
- Reports (панель звітів) - панель повідомлень: містить повідомлення, які відображають результати імітаційного моделювання;
- Navigate (панель навігації) - панель управління дозволяє відображати всі види моделі, включаючи управління через ієрархічні підмоделі.

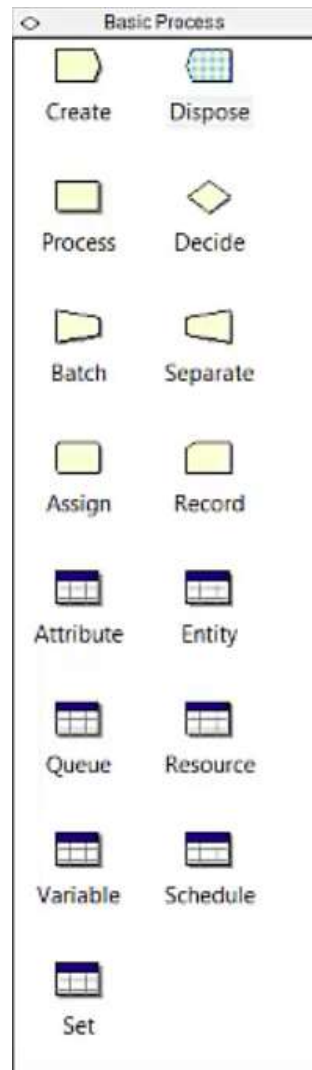
Вікно робочого поля представляє зображення моделі, включаючи блок-схему процесу, анімацію і інші елементи.

Вікно властивостей модуля використовуються для настройки параметрів моделі таких як: час, витрати і інші параметри.

Конструктивні елементи Arena

- Розділені на групи Basic Process, Advanced Process та інші
- Кожна група містить набір модулів для побудови моделі. Модулі перетягуються у вікно робочого поля та з'єднуються у відповідності до логіки процесу, який моделюється
- Кожен блок містить набір параметрів, які має налаштувати користувач у вікні властивостей модулю

Basic Process



Модуль *Create*

Модуль **Create** є відправною точкою для вимоги в імітаційній моделі. Вимоги - це об'єкти, які обробляються в системі. Створення вимоги модулем відбувається за розкладом, або ж ґрунтуючись на значенні часу між прибуттям вимог в модель. Залишаючи модуль, вимога починає оброблятися в системі. Тип створюваної вимоги визначається в цьому модулі. Параметри модуля та їх призначення:

- *Name* - унікальне ім'я модуля, яке буде відображено в блок схемі.
- *Entity Type* - назва типу вимоги, який створюватиметься модулем.
- *Type* - спосіб формування потоку прибуття. Type може мати значення Random (використовується експоненціальний розподіл з середнім значенням, визначеним користувачем), Schedule (визначається модулем Schedule), Constant (використовуватиметься, визначене користувачем, постійне значення) або Expression (потік прибуття формуватиметься по певному виразу).
- *Value* - визначає середнє значення експоненціального розподілу (Random) або постійне значення часу між прибуттям вимоги (якщо Type = Constant).
- *Schedule Name* - ім'я розкладу, який визначає характер прибуття вимоги в систему.
- *Expression* - задає тип розподілу або вираз, що визначає час між прибуттями вимоги в модель.
- *Units* - Одиниці вимірювання часу між прибуттям (день, година, хвилинилина, секунда).
- *Entities per arrival* - Кількість вимог, яка входить в систему за одне прибуття.
- *Max arrivals* - Максимальне число вимог, яке може створити цей модуль.
- *First Creation* - Час, через який прибуде перша вимога в модель від початку симуляції.

Модуль *Process*

Модуль *Process* є основним модулем процесу обробки в імітаційній моделі. Окрім стандартного модуля *Process*, можна використовувати підмодель, додаючи їй особливу, визначену користувачем, ієрархічну логічну схему. У модулі можна також задавати додаткові вартісні і часові характеристики процесу обробки об'єкта. Параметри модуля та їх значення:

- *Name* - унікальне ім'я модуля, яке відображається в блок схемі.
- *Type* - визначає логічну схему модуля. *Standard* означає, що логічна схема знаходиться усередині модуля і залежить від параметра *Action*. *Submodel* показує, що логічна схема знаходитиметься нижче в ієрархічній моделі. Підмодель може містити будь-яку кількість логічних модулів.
- *Action* - тип обробки що відбувається усередині модуля. *Delay* просто показує про те, що процес займає деякий час і не відображає використання ресурсів. *Seize Delay* указує на те, що в цьому модулі були розміщені ресурси і відбуватиметься затримка, ресурси захоплюватимуться (тобто будуть зайняті обробкою вимоги), і їх звільнення буде відбуватися пізніше. *Seize Delay Release* указує на те, що ресурс(-и) були захоплені, а потім через час звільнилися. *Delay Release* означає, що ресурси до цього були захоплені вимогою, а в такому модулі вимога затримується і звільнить ресурс. Всі ці параметри доступні тільки тоді, коли *Type* – *Standard*.

Модуль *Process*

- *Priority* - значення пріоритету модулів що використовують один і той же ресурс де завгодно в моделі. Ця властивість не доступна, якщо Action = Delay або Delay Release, або коли Type = Submodel.
- *Resources* - визначає ресурси або групи ресурсів, які оброблятимуть вимогу в цьому модулі (див. Модуль Process - Ресурси).
- *Delay Type* - тип розподілу або процедура, що визначає параметри затримки.
- *Units* - Одиниці вимірювання часу затримки (день, година, хвилинилина, секунда).
- *Allocation* - Визначає вартісні характеристики обробки. Value Added - означає враховувати вартісні характеристики, а Non-Value Added не враховувати.
- *Minimum* - мінімальне значення для рівномірного і трикутного розподілу.
- *Maximium* - максимальне значення для рівномірного і трикутного розподілу.
- *Value* - середнє значення для нормального і трикутного розподілу або значення для постійної часової затримки.
- *Std Dev* - стандартне відхилення для нормального розподілу.

Модуль *Decide*

Модуль Decide дозволяє враховувати прийняття рішень в моделі. Модуль включає опції прийняття рішень заснованих на умові By Condition (наприклад, якщо тип вимоги Car) або заснованих на вірогідності By Chance (наприклад, 75% - true, а 25% - false). Умови можуть бути засновані на значенні атрибуту Attribute, значенні змінної Variable, типі об'єкта Entity Type або засновані на виразі Expression. Якщо поставлена умова не виконується, то об'єкт залишатиме модуль через гілку False.

Параметри модуля та їх значення:

- Name - унікальне ім'я модуля, яке відображається в блок схемі.
- Type - тип ухвалення рішення. By Chance - вибір напряму ґрунтується на ймовірності. By Condition - перевірка на виконання умови
- Percent True - значення, що визначає відсоток вимог, який підуть по на-пряму True
- If - Тип умови, яка перевірятиметься на виконання.
- Named - ім'я змінної, атрибуту або типу вимоги, який перевірятимуться при вході вимоги в модуль.
- Is - математичний знак умови, наприклад більше, менше, рівно і т.д.
- Value - значення, з яким порівнюватиметься атрибут або змінна вимоги, що прийшла. Якщо тип умови - Expression, то у виразі повинен стояти знак умови, наприклад Color<> Red
- Даний модуль дозволяє виконувати перевірку не тільки однієї умови, але і декілька. Це досягається за допомогою властивості Type N-way by Chance/by Condition. Залежно від умови об'єкт йде по потрібній вітці.

Модуль *Batch*

Модуль ***Batch*** відповідає за механізм угруповання в імітаційній моделі. Наприклад, зібрати необхідну кількість даних, перш ніж починати їх обробку, зібрати раніше розділені копії одної форми, з'єднати пацієнта і його лікарняну карту прийому до лікаря. Угруповання може бути постійним або тимчасовим. Тимчасово згруповані комплекти пізніше можуть бути роз'єднані за допомогою модуля *Separate*. Комплекти можуть складатися з будь-якої кількості вхідних об'єктів, визначеної користувачем, або ж об'єкт може об'єднуватися в комплект залежно від його атрибуту. Часові і вартісні характеристики об'єктів, які входять у групу, будуть рівні сумі характеристик вимог, які увійшли до групи.

- вимога прибуває в модуль, стає в чергу і залишається там до тих пір, поки в модулі не буде набрано задану кількість вимог. Коли набереться потрібне число об'єктів створюється об'єкт, який представляє комплект. Параметри модуля та їх значення:
- *Name* - унікальне ім'я модуля, яке відображається в блок схемі.
- *Type* - Спосіб угруповання вимоги, може бути *Temporary* (тимчасова), *Permanent* (постійна).
- *Batch Size* - Число вимог, які утворюють один комплект.
- *Rule* - Визначає, за якою ознакою групуватимуться. Якщо *Rule* = *Any Entity*, це означає що перші 3 (якщо *Batch Size* = 3) вимоги будуть згруповано. Якщо *Rule* = *By Attribute*, то об'єднуватиметься задана кількість вимог з певним атрибутом. Наприклад, якщо *Attribute Name* = *Color*, то всі вимоги, які мають однаковий атрибут *Color*, буде згрупована.
- *Attribute Name* - Ім'я атрибуту, по значенню якого групуватимуться вимоги.

Модуль *Separate*

Модуль ***Separate*** може використовуватися як для створення копій вхідної вимоги, так і для розділення раніше згрупованих вимог. Наприклад, для роз'єднання раніше згрупованих комплектів документів, для паралельної обробки рахунків і документів по одному замовленню. Правило для розділення вартісних і часових характеристик копій вимоги і розділеної вимоги визначається користувачем. Коли тимгодино згрупована вимога прибуває в модуль, вони розкладаються на складену вимогу. вимога покидає модуль в тій же послідовності, в якій вони додавалися в комплект. Якщо модуль створює копії вимоги, то користувач може задати кількість дублікатів вимоги. У дубльованій вимоги значення атрибуту, а також анімаційна картинка такі ж, як і оригіналу. Оригінальна вимога також покидає модуль. Параметри модуля та їх значення:

- *Name* - унікальне ім'я модуля, яке відображається в блок схемі.
- *# of Duplic* - Кількість створюваних копій вхідної вимоги.
- *Type* - Спосіб розділення вхідної в модуль вимоги. *Duplicate Original* - просто робить дублікати вхідної вимоги. *Split Existing Batch* вимагає щоб вхідна вимога була заздалегідь тимгодино згрупована.
- *Cost to Duplicates* - Розділення вартісних і часових характеристик вхідної вимоги між тими, що виходять. Це значення визначається користувачем у відсотках, тобто скільки відсотків від вартісних і часових характеристик вхідної вимоги піде копіям (характеристики між копіями діляться порівну).
- *Allocation Rule* - Метод розділення вартості і часу, якщо вибраний *Type=Split Existing Batch*. *Retain Original Entity Values* - зберігає оригінальні значення вимоги. *Take All Representative Values* - вся вимога приймає однакове значення. *Take Specific Representative Values* - вимога приймає специфічне значення.

Модуль *Assign*

Модуль ***Assign*** призначений для завдання нового значення змінній, атрибуту вимоги типу вимоги, анімаційній картинці вимоги або іншої змінної в системі. Наприклад, для встановлення пріоритету для клієнтів, для привласнення номера наказу, що вийшов. У одному модулі можна зробити тільки одне призначення. Параметри модуля та їх значення:

- *Name* - унікальне ім'я модуля, яке відображається в блок схемі.
- *Type* - тип призначення, яке здійснюватиметься. *Other* може включати вбудовані в Арену змінні, такі як місткість ресурсу або кінцевий час симуляції.
- *Variable Name* - ім'я змінної, яка змінюватиметься в цьому модулі.
- *Attribute Name* - ім'я атрибуту, який змінюватиметься в цьому модулі.
- *Entity Type* - новий тип вимоги, який привласнюється вимозі в цьому модулі.
- *Entity Picture* - нова анімаційна картинка для вимоги, яка пройшла цей модуль.
- *Other* - ім'я змінної в системі, яка змінюється.
- *New Value* - привласнюване нове значення для атрибуту, змінної.
-

Модуль *Record*

Модуль ***Record*** призначений для збору статистики в імітаційній моделі. Наприклад, для підрахування, яка кількість замовлень була виконана із запізненням, підрахування кількості роботи, що здійснюється за одну годину. Модуль може збирати різні типи статистики, включаючи час між виходами вимоги з модуля, статистику вимоги (час циклу, вартість), статистику за період часу (період часу від заданої точки до теперішнього моменту). Також доступний кількісний тип статистики. Параметри модуля та їх значення:

- *Name* - унікальне ім'я модуля, яке відображається в блок схемі.
- *Type* - визначає тип статистики, яка збиратиметься. *Count* - збільшуватиме або зменшуватиме статистику на задане значення. *Entity Statistics* збиратиме загальну статистику про вимогу, наприклад, час циклу, вартісні характеристики і т.д. *Time Interval* рахуватиме різницю між значенням атрибуту і поточним часом моделювання. *Time Between* відстежуватиме час між входженням вимоги в модуль. *Expression* просто фіксуватиме значення визначуване виразом.
- *Attribute Name* - ім'я атрибуту, значення якого використовуватиметься для інтервальної статистики.
- *Value* - значення, яке додаватиметься до статистики, коли в модуль прибуватиме вимога.

Модуль *Dispose*

Модуль *Dispose* є вихідною точкою з імітаційної моделі. Наприклад, клієнти покидають відділ, закінчення бізнес-процесу. Статистика про вимогу може збиратися до того моменту поки вона не вийде з системи. Параметри модуля та їх значення:

- *Name* - унікальне ім'я модуля, яке відображається в блок схемі.
- *Record Entity Statistics* - визначає, чи вести́меться статистика про вихід вимоги з системи.

Виконання моделювання

Перед тим, як розпочати процес імітації потрібно зайти в меню Run, підменю Setup і ввести в установки Replication Parameters час моделювання (Replication Length) та одиниці, в яких він вимірюється (Time Units), а також одиниці часу, в яких будуть представлені результати моделювання (Basic Time Units). Бажано також ввести в установки Project Parameters ім'я проекту (Project Title) обрати статистичні дані, що збираються (Entities, Queues, Resources, Processes, а також Costing).

Запуск імітації здійснюється вибором команди Go меню Run. Для перевірки моделі зручно скористатись командою Step, що запускає покрокову імітацію модельованого процесу. Управління процесом імітації здійснюють команди Pause - часова зупинка процесу імітації, Fast Forward - швидка імітація, End - повернення до корегування моделі або нового запуску.

Виведення результатів моделювання

- В результаті успішної імітації моделі пакетом Arena створюються звіти про результати моделювання. Переглянути їх можна скориставшись панеллю звітів *Reports*. На панелі звітів представлено декілька видів звітів. Звіт «Короткий огляд категорій» і звіти по чотирьох категоріях, такі як Вимоги, Процеси, Черги і Ресурси.
- За допомогою панелі звітів можна переглянути звіт *Category Overview*, який відображає підсумкову інформацію про вимоги, процеси, черги і ресурси. Також показує інформацію про заданих користувачем змінних і інформацію, зібрану модулем *Record*.
- Звіт про вимоги *Entities* поділений на декілька частин. У частині звіту *Cycle Time* показаний середній, максимальний і мінімальний час існування вимоги. Час існування вимоги вважається з моменту її прибуття в систему і до того моменту, коли вимога потрапляє в модуль *Dispose*. Нижче представляється гістограма середнього часу циклу для кожного типу вимоги.

Приклад

Експериментальна роботизована гнучка виробнича система має два верстати із числовим пультом керування, три роботи, пункт прибуття і склад оброблених деталей. Деталі прибувають на пункт прибуття кожні 40 секунд згідно з експоненціальним законом розподілу, захоплюються одним з вільних роботів і переміщуються ним до першого верстата, після чого робот звільняється. Після завершення обробки на першому верстаті деталь захоплюється одним з роботів і переміщується на другий верстат, а після обробки на другому верстаті – одним з роботів переміщується на склад оброблених деталей. Кожний з верстатів може одночасно обробляти до трьох деталей.

Час переміщення робота між пунктом прибуття і першим верстатом, першим і другим верстатом, другим верстатом і пунктом зберігання оброблених деталей складає відповідно 6, 7, і 5 секунд незалежно від того, холостий це хід, чи ні. Роботу потрібний час 8 ± 1 секунд на захоплення або вивільнення деталей. Час обробки на першому верстаті розподілений за нормальним законом із середнім значення 60 секунд і має стандартне відхилення 10 секунд. Середній час обробки на другому верстаті дорівнює 100 секунд і має експоненціальний закон розподілу.

Визначить найкращий (з точки зору підвищення пропускної здатності гнучкої виробничої системи) спосіб закріплення роботів до операцій. Можливі варіанти закріплення:

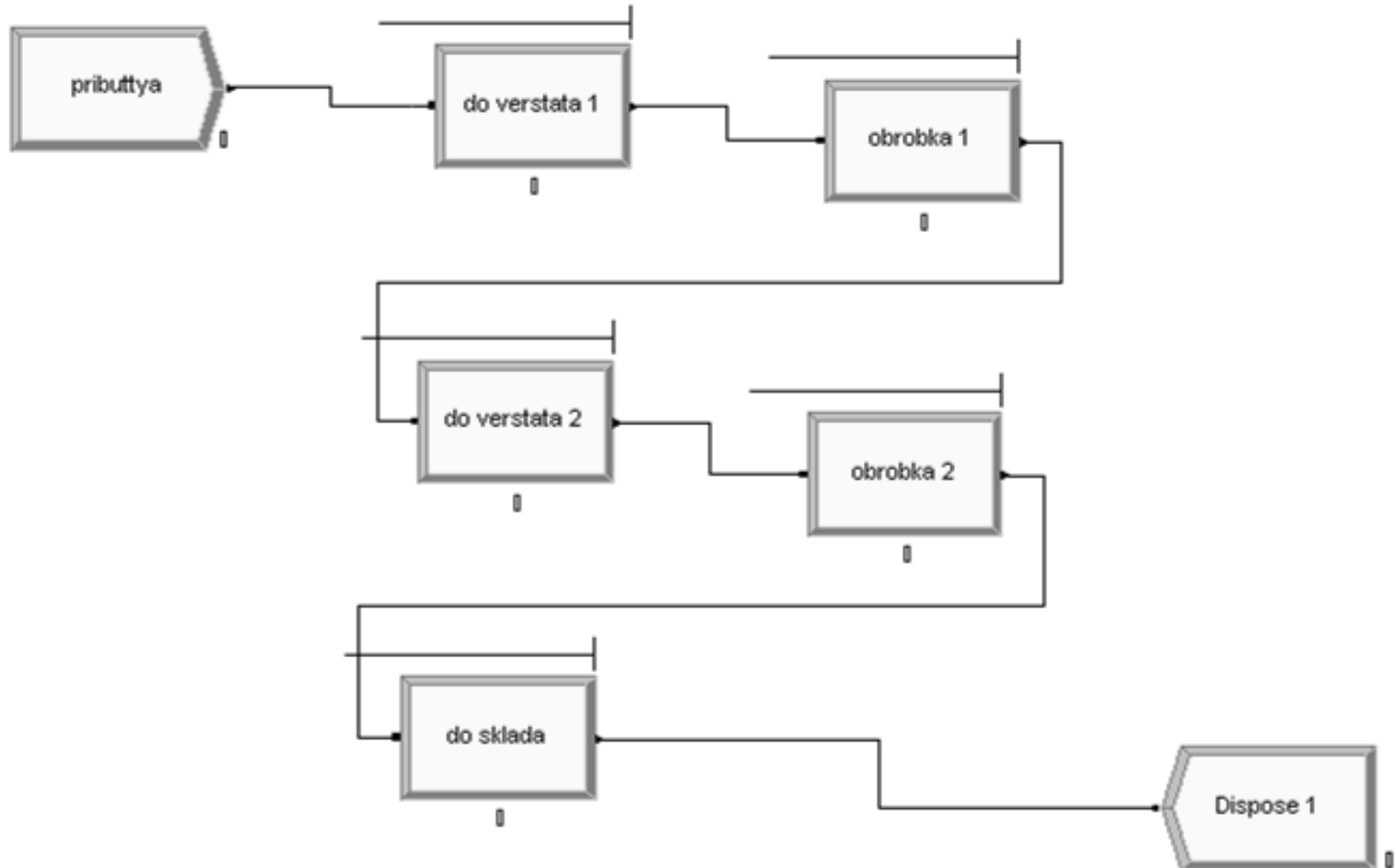
1. по одному роботу на кожний з трьох шляхів переміщення деталей (пункт прибуття – перший верстат, перший верстат – другий верстат, другий верстат, склад);
2. кожний робот може використовуватися на кожному шляху переміщення деталей (при цьому повинен займатися найближчий з роботів).

Знайдіть коефіцієнти використання роботів і верстатів, максимальну місткість місця зберігання деталей на ділянці прибуття.

Пояснення до моделі

- Обробку деталі першим верстатом моделює блок Process з параметрами Name=obrobka 1, Type=Standard, Action=Seize Delay Release, Resources=verstat 1, Delay Type=Normal, Units=Seconds , Value=60, Std Dev=10. Щоб верстат обслуговував одночасно три деталі, потрібно зайти в блок Resources і встановити параметр Capacity=3 для ресурсу з ім'ям verstat 1.
- Захоплення деталі роботом, переміщення її до другого верстата та вивільнення робота моделює блок Process з параметрами Name=do verstata 1, Type=Standard, Action=Seize Delay Release, Resources= Robot 2, Delay Type=Uniform, Units=Seconds , Minimum=22, Maximum=24.
- Змінемо закріплення роботів до операцій так, що кожний з вільних роботів може використовуватися на кожному шляху переміщення деталей. Для цього в моделі, яка складена, параметр Resources для процесів do verstata 1, do verstata 1, do sklada, потрібно задати як множина ресурсів, тобто тип Set, з ім'ям Set Robot. Потім за допомогою блоку Set задати елементи множини ресурсів Robot 1, Robot 2, Robot 3.

Модель роботизованої системи



Результати моделювання

1-ий варіант

- кількість оброблених деталей – 716 деталей;
- середнє завантаження першого верстата – 0,51;
- середня кількість деталей на обробці на першому верстаті – 1,52;
- середнє завантаження другого верстата – 0,81;
- середня кількість деталей на обробці на другому верстаті – 2,43;
- середнє завантаження першого робота – 0,56;
- середнє завантаження другого робота – 0,55;
- середнє завантаження третього робота – 0,55;
- середній час обробки однієї деталі – 314 с;
- середній час очікування однієї деталі – 90с, з них в очікуванні вільного робота – 6с, в очікуванні вільного першого верстата – 4с, в очікуванні вільного другого верстата – 80с;

Результати моделювання

2-ий варіант

- кількість оброблених деталей – 745 деталей;
- середнє завантаження першого верстата – 0,52;
- середня кількість деталей на обробці на першому верстаті – 1,56;
- середнє завантаження другого верстата – 0,80;
- середня кількість деталей на обробці на першому верстаті – 2,40;
- середнє завантаження першого робота – 0,57;
- середнє завантаження другого робота – 0,60;
- середнє завантаження третього робота – 0,54;
- середній час обробки однієї деталі – 356 с;
- середній час очікування однієї деталі – 137с, з них в очікуванні вільного робота – 35с, в очікуванні вільного першого верстата – 1с, в очікуванні вільного другого верстата – 101с.