Symulacja cyfrowa
Projekt

Alan Czyżewski

Metoda symulacji:
Planowanie zdarzeń

1. Treść zadania

W chińskiej restauracji pracuje k kelnerów obsługujących n_2 stolików dwuosobowych, n_3 stolików trzyosobowych oraz n_4 stolików czteroosobowych. Klienci pojawiają się w restauracji jako grupy 1-, 2-, 3- lub 4-osobowe z prawdopodobieństwami odpowiednio p1, p2, p3 oraz p4. Odstęp czasu rozdzielający pojawienie się kolejnych grup klientów jest zmienną losową o rozkładzie normalnym ze średnią μ_a i wariancją σ_a^2. Jeśli jest dostępny stolik odpowiadający wielkości grupy (lub większy), klienci są do niego prowadzeni przez kierownika sali (czynność ta zajmuje s jednostek czasu). W przeciwnym przypadku grupa oczekuje na stolik w kolejce. Średnio 50% klientów korzysta z samoobsługowego bufetu, przy którym może znajdować się jednocześnie b osób. Czas spędzany przy bufecie przez grupę klientów jest zmienną losową o rozkładzie normalnym ze średnią μ_b i wariancją σ_b^2. Pozostali klienci są obsługiwani przez tego z kelnerów, który jako pierwszy będzie wolny. W pierwszej kolejności klienci otrzymują napoje, a następnie serwowane jest danie główne. Czas obsługi w obu przypadkach jest zmienną losową o rozkładzie wykładniczym ze średnimi odpowiednio λ_n oraz λ_j (te dwie wielkości uwzględniają zarówno czas oczekiwania na zrealizowanie zamówienia jak i sam czas podania napojów i posiłku głównego). Po zakończeniu konsumpcji, której długość jest zmienną losową o rozkładzie wykładniczym ze średnią λ_f, klient płaci jednemu z c zatrudnionych kasjerów. Czas obsługi przez kasjera jest zmienną losową o rozkładzie wykładniczym ze średnią λ_p.

a) średni czas oczekiwania na stolik,

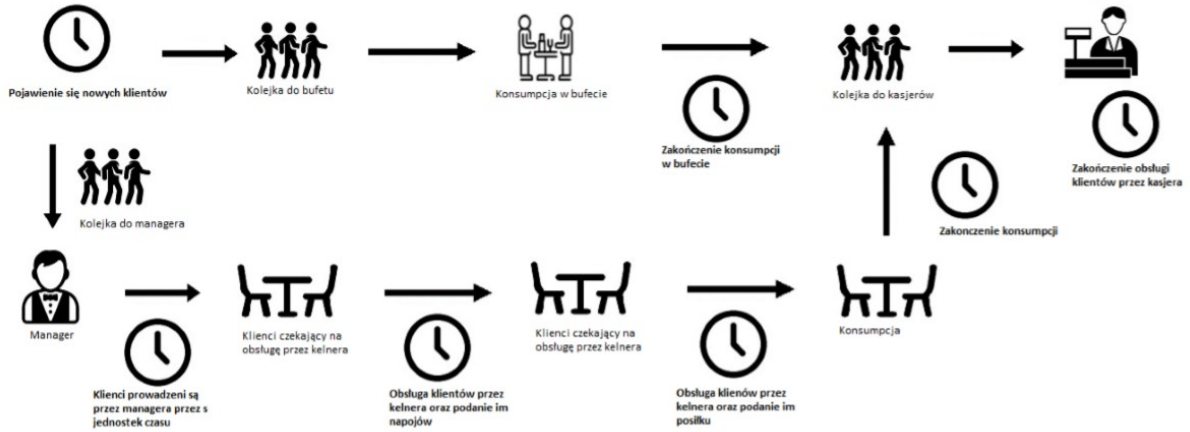
b) średnią długość kolejki oczekujących na stolik,

c) średni czas oczekiwania na obsługę przez kelnera od momentu zajęcia miejsca przy stoliku,

d) średnią długość kolejki przy kasach.

2. Opis modelu symulacyjnego

2.1. Schemat modelu symulacyjnego.



2.2. Opis klas wchodzących w skład systemu i ich atrybutów.

Nazwa klasy	**Opis**	**Atrybuty**
Restaurant	Klasa reprezentująca restaurację. Zawarte w niej są niezbędne obiekty i metody działające na tych obiektach.	- inteligentny wskaźnik na listę zdarzeń typu *std::shared_ptr<EventList>* - inteligentny wskaźnik na statystyki typu *std::shared_ptr<Statistics>* - aktualny czas symulacji typu *double* - lista klientów znajdujących się aktualnie w restauracji typu *std::list<std::shared_ptr<Client>>* - kolejka klientów do bufetu typu *std::queue<std::shared_ptr<Client>>* - kolejka klientów do kasy typu *std::queue<std::shared_ptr<Client>>* - kolejka klientów oczekujących na stolik typu *std::queue<std::shared_ptr<Client>>* - kolejka klientów oczekujących na kelnera typu *std::queue<std::shared_ptr<Client>>* - stała zawierająca liczbę miejsc przy bufecie typu *const int* -stała zawierająca liczbę stolików 2-osobowych typu *const int* -stała zawierająca liczbę stolików 3-osobowych typu *const int* -stała zawierająca liczbę stolików 4-osobowych typu *const int*

		- stała zawierająca liczbę kelnerów typu *const int* - stała zawierająca liczbę kasjerów typu *const int* - zmienna reprezentująca liczbę wolnych miejsc przy bufecie typu *int* - kontener asocjacyjny map zawierający liczbę wolnych stolików różnego rodzaju typu *std::map<const int, int>* - zmienna reprezentująca liczbę wolnych kelnerów typu *int* - zmienna reprezentująca liczbę wolnych kasjerów typu *int* -zmienna mówiąca o tym, czy kierownik nie jest zajęty typu *bool*
Client	Klasa reprezentująca grupę klientów (1-, 2-, 3- lub 4-osobową)	- zmienna oznaczająca identyfikator, który jest przypisany nowej grupie klientów typu *static int* - stała reprezentująca identyfikator grupy klientów typu *const int* - zmienna reprezentująca liczbę osób w grupie typu *int* - zmienna reprezentująca liczbę miejsc przy stoliku zajmowanym przez grupę typu *int* - zmienna reprezentująca liczbę miejsc przy drugim stoliku zajmowanym przez grupę typu *int* - zmienna reprezentująca czas spędzony przy bufecie typu *double* - zmienna reprezentująca czas podawania napojów tej grupie typu *double* - zmienna reprezentująca czas podawania jedzenia tej grupie typu *double* - zmienna reprezentująca czas konsumpcji grupy typu *double* - zmienna reprezentująca czas spędzony przy kasie typu *double* - zmienna mówiąca o tym, czy zostały już podane napoje tej grupie typu *bool* - zmienna reprezentująca czas rozpoczęcia czekania na stolik typu double - zmienna reprezentująca czas zajęcia stolika typu double

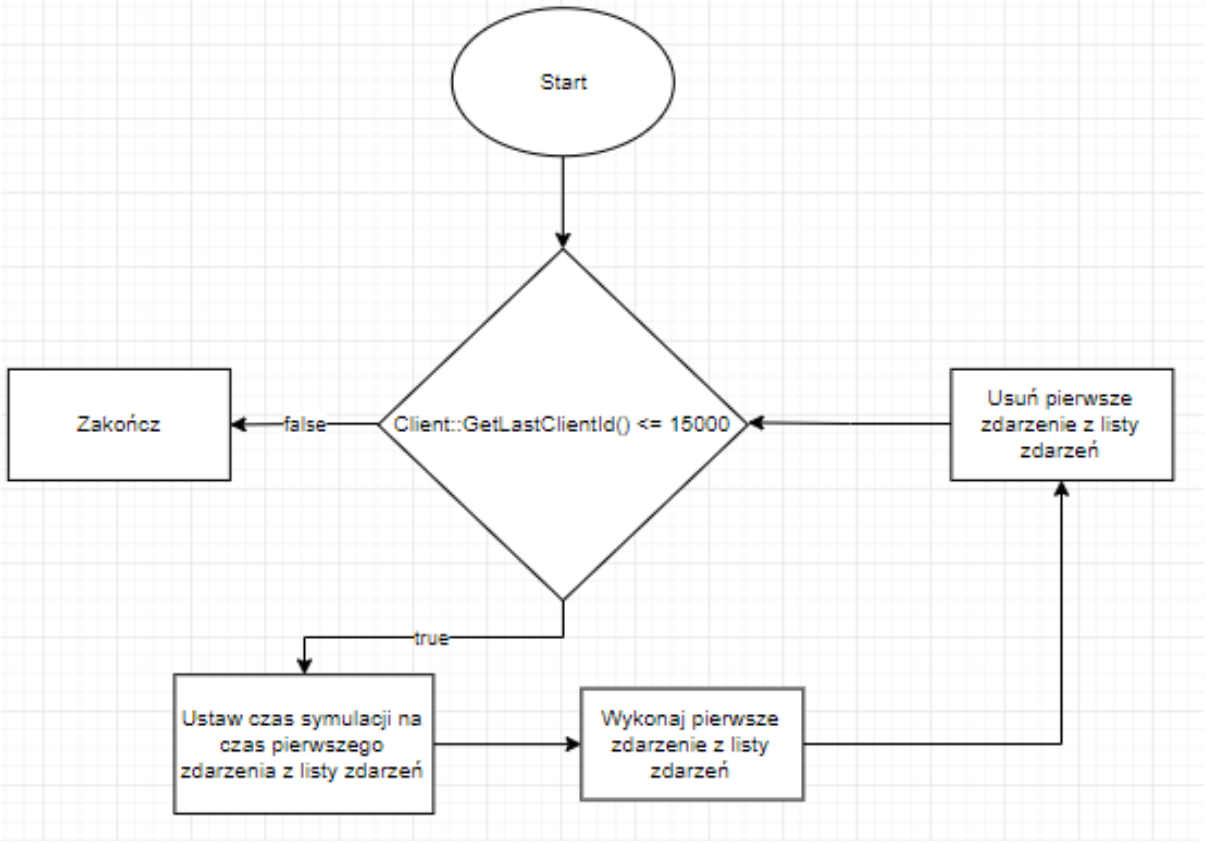
Event	Klasa reprezentująca zdarzenie czasowe.	-zmienna zawierająca nazwę zdarzenia typu *std::string* - zmienna zawierająca czas zdarzenia typu *double* - wskaźnik na restaurację, dla której wykonywana jest symulacja typu *Restaurant** - inteligenty wskaźnik na grupę klientów związanych ze zdarzeniem typu *std::shared_ptr<Client>*
EventList	Klasa reprezentująca listę zdarzeń	- lista zdarzeń typu *std::list<std::shared_ptr<Event>>*
ClientArrival	Klasa reprezentująca zdarzenie czasowe - pojawienia się nowego klienta.	- wszystkie atrybuty znajdują się w klasie bazowej *Event*
BuffetServiceEnd	Klasa reprezentująca zdarzenie czasowe – zakończenie usługi bufetu.	- wszystkie atrybuty znajdują się w klasie bazowej *Event*
TableTaken	Klasa reprezentująca zdarzenie czasowe – posadzenie grupy do stolika.	- wszystkie atrybuty znajdują się w klasie bazowej *Event*
ServedDrinks	Klasa reprezentująca zdarzenie czasowe – podanie napojów.	- wszystkie atrybuty znajdują się w klasie bazowej *Event*
ServedFood	Klasa reprezentująca zdarzenie czasowe – podanie jedzenia.	- wszystkie atrybuty znajdują się w klasie bazowej *Event*

ConsumptionEnd	Klasa reprezentująca zdarzenie czasowe – zakończenie konsumpcji.	- wszystkie atrybuty znajdują się w klasie bazowej *Event*
CheckoutServiceEnd	Klasa reprezentująca zdarzenie czasowe – zakończenie obsługi przez kasjera.	- wszystkie atrybuty znajdują się w klasie bazowej *Event*
Statistics	Klasa zawierająca statystyki symulacji.	-zmienne używane do obliczenia średniego czasu oczekiwania na stolik -zmienne używane do obliczenia średniej długości kolejki do stolika -zmienne używane do obliczenia średniego czasu oczekiwania na obsługę przez kelnera - zmienne używane do obliczenia średniej długości kolejki przy kasach
namespace Generators	Przestrzeń nazw zawierająca generatory liczb pseudolosowych.	- zmienna reprezentująca ziarno potrzebne do generowania kolejnych liczb pseudolosowych typu *static int* - statyczne zmienne z treści zadania, służące do zainicjalizowania wartości danych: k, n2, n3, n4, p1, p2, p3, p4, mi_a, sigma_a, s, b, mi_b, sigma_b, lambda_n, lambda_j, lambda_f, c, lambda_p, mi_e, sigma_e

3. Opis metody symulacyjnej.

Metoda: Planowanie zdarzeń.

3.1. Schemat blokowy pętli głównej.



3.2. Lista zdarzeń czasowych.

Zdarzenia warunkowe zawarte są w zdarzeniach czasowych.

Zdarzenia czasowe:

1. Pojawienie się nowego klienta.

- Dodaj grupę do systemu.

- Skieruj grupę do bufetu (45%) lub do stolika (55%). Jeśli nie ma odpowiedniej liczby miejsc, ustaw grupę w kolejce.

- Zaplanuj kolejne pojawienie się nowego klienta.

2. Zakończenie usługi bufetu.

- Usuń grupę z bufetu (zwolnij tyle miejsc, ile zajmowała grupa).

- Jeśli przy bufecie jest wystarczająca liczba miejsc, umieść w niej pierwszą grupę z kolejki. Rób to do momentu, gdy pierwsza grupa z kolejki nie zmieści się w bufecie.

- Jeśli jest wolna kasa, umieść w niej grupę i zaplanuj zdarzenie zakończenia obsługi przez kasjera. W przeciwnym razie umieść ją w kolejce do kasy.

3. Posadzenie grupy do stolika.

- Zwolnij managera i zaplanuj posadzenie kolejnej grupy do stolika (pierwszej z kolejki), jeśli są wolne miejsca.

- Jeśli nie ma wolnego kelnera, umieść grupę w kolejce do kelnera. W przeciwnym razie zaplanuj zdarzenie zakończenia podania napojów.

4. Podanie napojów.

- Umieść grupę na końcu kolejki oczekujących na obsługę kelnera.

- Zwolnij kelnera.

- Wybierz z kolejki do kelnera pierwszą grupę. Jeśli grupa dostała już napoje zaplanuj dla niej zakończenie podania jedzenia. W przeciwnym razie zaplanuj zakończenie podania napojów.

5. Podanie jedzenia.

- Zaplanuj grupie zakończenie konsumpcji.

- Zwolnij kelnera.

- Wybierz z kolejki do kelnera pierwszą grupę. Jeśli grupa dostała już napoje zaplanuj dla niej zakończenie podania jedzenia. W przeciwnym razie zaplanuj zakończenie podania napojów.

6. Zakończenie konsumpcji.

- Zwolnij stolik(i) przypisany(e) grupie.

- Zaplanuj posadzenie kolejnej grupy do stolika (pierwszej z kolejki), jeśli są wolne miejsca.

- Jeśli jest wolna kasa, umieść w niej grupę i zaplanuj zdarzenie zakończenia obsługi przez kasjera. W przeciwnym razie umieść ją w kolejce do kasy.

7. Zakończenie obsługi przez kasjera.

- Usuń klienta z systemu.

- Zwolnij miejsce przy kasie.

- Jeśli kolejka do kasy nie jest pusta, zaplanuj zakończenie obsługi przez kasjera pierwszej grupie z kolejki.

4. Parametry wywołania programu.

```
static const std::array<int, 10> seeds = { 63527281, 738291392, 612039133, 129348237, 425163492,
                                           22248329, 65428821, 773629001, 737269203, 1002363281 };
static int seed = seeds[0];
static int p = 0.45; //probability (buffet)
static int n2 = 4; //number of 2-person tables
static int n3 = 10; //number of 3-person tables
static int n4 = 4; //number of 4-person tables
static int b = 14; //seats at the buffet
static int mi_b = 3200; //time buffet
static int lambda_f = 1900; //time consumption
static int c = 4; //number of checkouts
static int w = 7; //number of waiters
static double p1 = 0.15; //probability (1-person)
static double p2 = 0.36; //probability (2-person)
static double p3 = 0.30; //probability (3-person)
static double p4 = 0.19; //probability (4-person)
static int mi_a = 330; //arrival new clients
static int sigma_a = 20; //arrival new clients
static int s = 40; //time manager
static int sigma_b = 100; //time buffet
static int lambda_n = 1100; //time serving drinks
static int lambda_j = 2400; //time serving food
static int lambda_p = 700; //time checkout
```

5. Generatory.

W programie zastosowałem 3 generatory liczb pseudolosowych: o rozkładzie równomiernym, o rozkładzie wykładniczym i o rozkładzie normalnym. Dwa ostatnie w swojej implementacji zawierają generator o rozkładzie równomiernym.

Ziarna użyte w symulacji są losowe.

Ziarna początkowe użyte w 10 niezależnych symulacjach:

63527281	738291392	612039133	129348237	425163492
22248329	65428821	773629001	737269203	1002363281

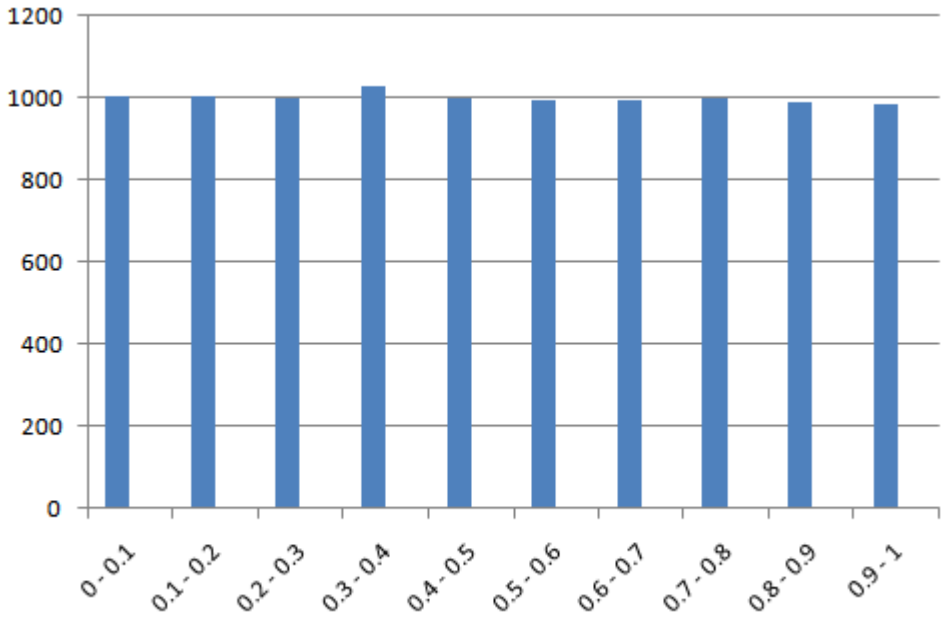
W trakcie generowania każdej liczby pseudolosowej o rozkładzie równomiernym następuje zmiana wartości ziarna.

Generator liczb pseudolosowych o rozkładzie równomiernym

Poniżej znajduje się kod generatora:

```
double Uniform()
{
    const double m = 2147483647;
    const auto a = 16807;
    const auto q = 127773;
    const auto r = 2836;
    auto h = seed / q;
    seed = a * (seed - q * h) - r * h;
    if (seed < 0) seed += m;
    return seed / m;
}
```

Histogram:

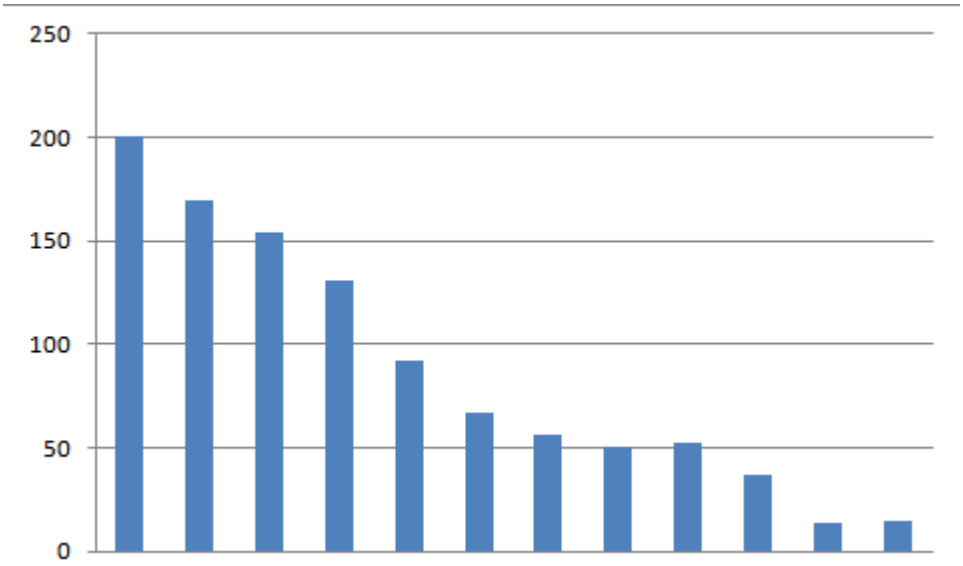


Generator liczb pseudolosowych o rozkładzie wykładniczym

Poniżej znajduje się kod generatora:

```
double Exponential(const double avg)
{
    return (-1)*(log(Uniform()))*avg;
}
```

Histogram:



Generator liczb pseudolosowych o rozkładzie normalnym

Poniżej znajduje się kod generatora:

```
double Normal(const int average, const double variance)
{
    double x = 0;
    const auto n = 12;
    for (auto i = 0; i < n; i++)
        x += Uniform();
    return ((x - (double(n) / 2))*(variance * 12 / n)) + average;
}
```

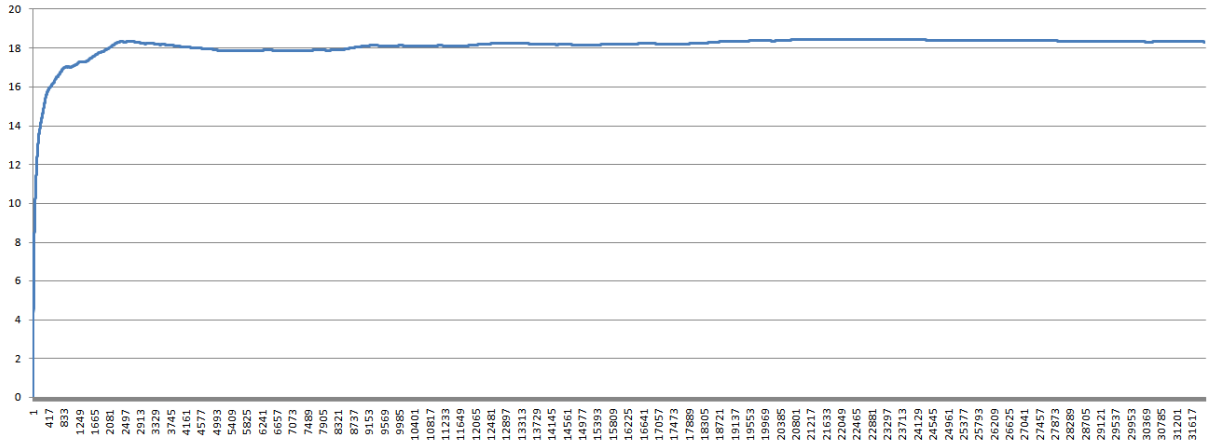
Histogram:

6. Opis zastosowanej metody testowania i weryfikacji poprawności działania programu.

Weryfikacja poprawności działania programu polegała głównie na używaniu trybu krokowego i wyświetlaniu aktualnego stanu symulacji. W przypadku jakichkolwiek niejasności i błędów w kodzie używałem debuggera do krokowego przechodzenia po instrukcjach programu.

7. Wyniki symulacji

7.1. Wyznaczenie długości fazy początkowej na podstawie średniej liczby klientów



Długość fazy początkowej bezpiecznie ustaliłem na 8000 klientów.

7.2. Wyznaczenie parametrów

Parametry podane w zadaniu nie dawały satysfakcjonujących rezultatów. Z tego względu należało zmienić niektóre z nich.

Parametr	**Znaczenie parametru**	**Wartość startowa**	**Wartość końcowa**
p	Prawdopodobieństwo wybrania bufetu	**0.5**	0.45
k	Liczba kelnerów	**7**	7
n_2, n_3, n_4	Liczba stolików (2-, 3- i 4-osobowych)	**4,10,4**	4,10,4
p_1, p_2, p_3, p_4	Prawdopodobieństwo przyjścia grupy(1-, 2-, 3- i 4-osobowej)	0.11, 0.33, 0.33, 0.23	**0.15, 0.36, 0.30, 0.19**
$\mu_a(\sigma_a^2)$	Odstęp czasu pojawiania się następnych klientów	1900 (200^2)	**330**(20^2)
s	Czas prowadzenia grupy przez managera	30	**40**
b	Liczba miejsc w bufecie	**14**	14
$\mu_b(\sigma_b^2)$	Czas spędzony przy bufecie	**3200** (100^2)	3200(100^2)
λ_n	Czas podawania napojów	300	**1100**
λ_j	Czas podawania posiłków	1700	**2400**

λ_f	Czas konsumpcji	**1900**	1900
c	Liczba kasjerów	**4**	4
λ_p	Czas spędzony przy kasie	200	**700**

7.3. Wyniki symulacji dla każdego przebiegu symulacyjnego.

Tabela z wynikami:

Średni czas oczekiwania na stolik	Średni czas oczekiwania na obsługę przez kelnera	Średnia długość kolejki do kasy	Średnia długość kolejki do stolika
101.55	412.63	0.14230	0.17187
92.99	376.97	0.15451	0.15368
144.59	334.42	0.14097	0.23923
105.34	469.32	0.15064	0.17814
180.34	452.17	0.13477	0.30016
120.14	358.94	0.18077	0.19843
209.83	454.56	0.14825	0.35030
68.37	331.35	0.13481	0.11428
76.66	353.59	0.13938	0.12727
69.82	326.52	0.16112	0.11480

Powyższe wyniki pochodzą z 10 symulacji z 10 różnymi ziarnami.

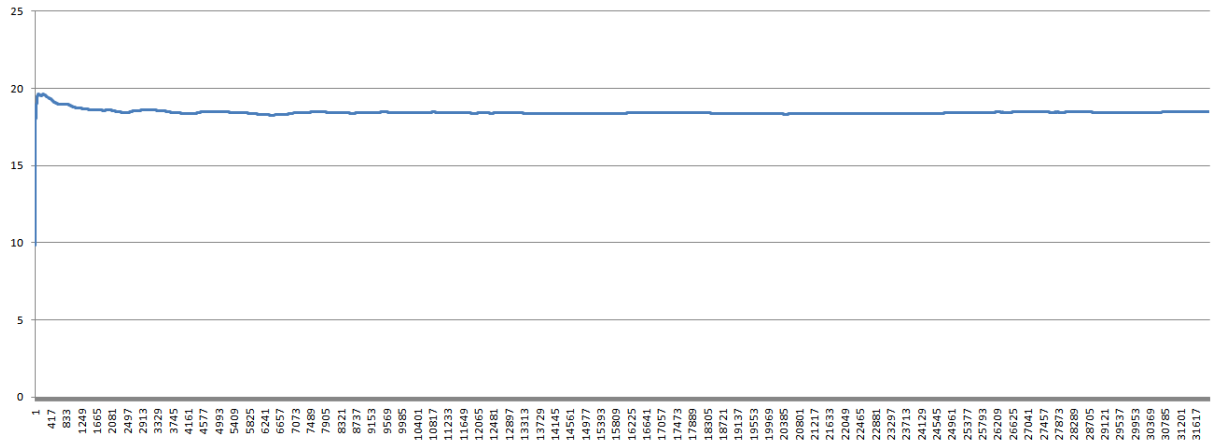
Uśrednione wyniki z 10 symulacji:

Średni czas oczekiwania na stolik	Średni czas oczekiwania na obsługę przez kelnera	Średnia długość kolejki do kasy	Średnia długość kolejki do stolika
116.96	387.05	0.1487	0.1948

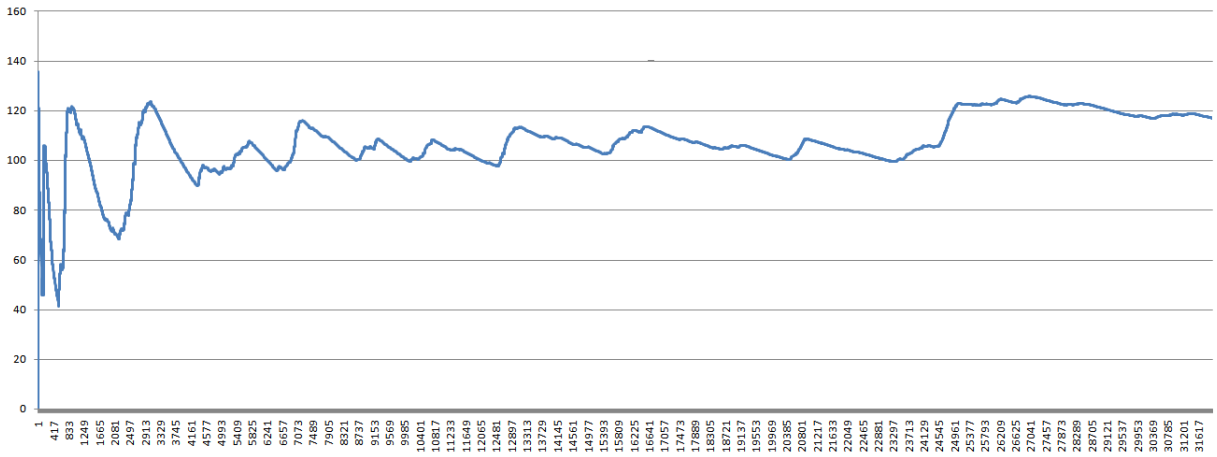
7.4. Wyniki końcowe.

Poniższe wykresy i wyniki zostały uzyskane po odcięciu fazy początkowej i uśrednieniu wyników 10 niezależnych symulacji.

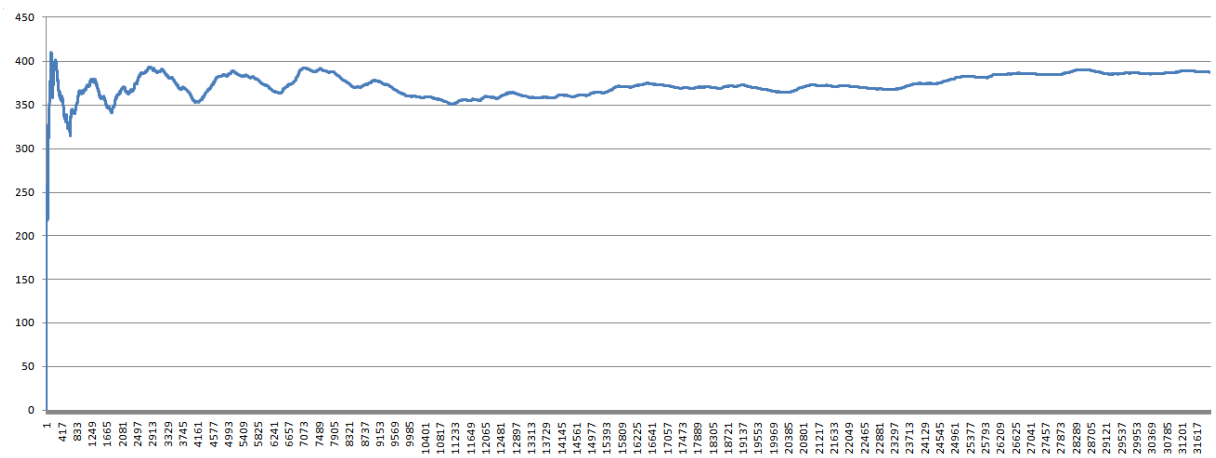
Średnia liczba klientów



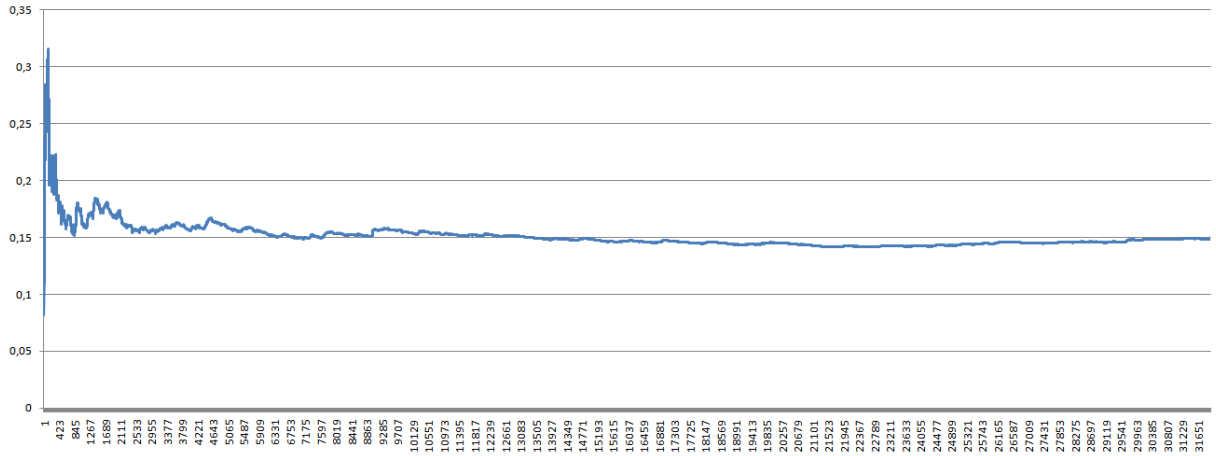
Średni czas oczekiwania na stolik:



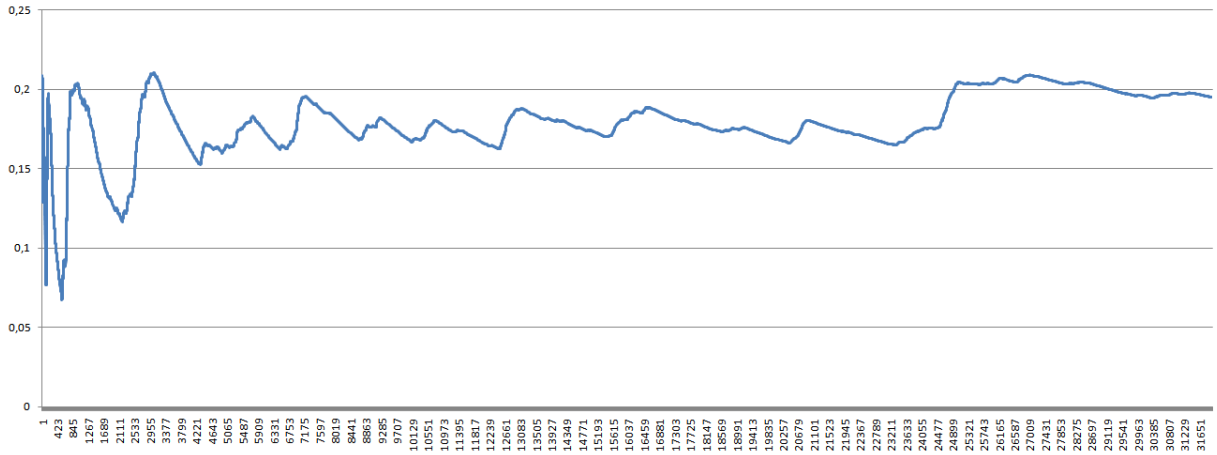
Średni czas oczekiwania na obsługę przez kelnera:



Średnia długość kolejki do kasy:



Średnia długość kolejki do stolika:



Wyniki końcowe:

Parametr	**minimum**	**średnia**	**Maksimum**
średni czas oczekiwania na stolik	87.33237	116.96	146.5936
średni czas oczekiwania na obsługę kelnera	352.6197	387.05	421.4743
średnia długość kolejki do kasy	0.1399	0.1487	0.1575
średnia długość kolejki do stolika	0.1454	0.1948	0.2442

W powyższych wynikach zastosowano przedział ufności 95%

8. Wnioski

Symulacja przebiegła sprawnie, ale trudno było dobrać właściwe parametry tak, aby system się nie przepełniał klientami, a zarazem aby ich nie brakowało. Metoda planowania zdarzeń mimo trywialnej pętli symulacyjnej, wymagała poświęcenia wiele czasu między innymi na zastanowienie się jakie warunki należy sprawdzić, przed zakończeniem i usunięciem zdarzenia czasowego. Wykresy zawarte w tym raporcie pokazują, że system symulacyjny dąży do ustabilizowania się.