

Sprawozdanie: Testy systemu transputerowego

Szymon Francuzik Stanisław Jankowski

24 listopada 2008

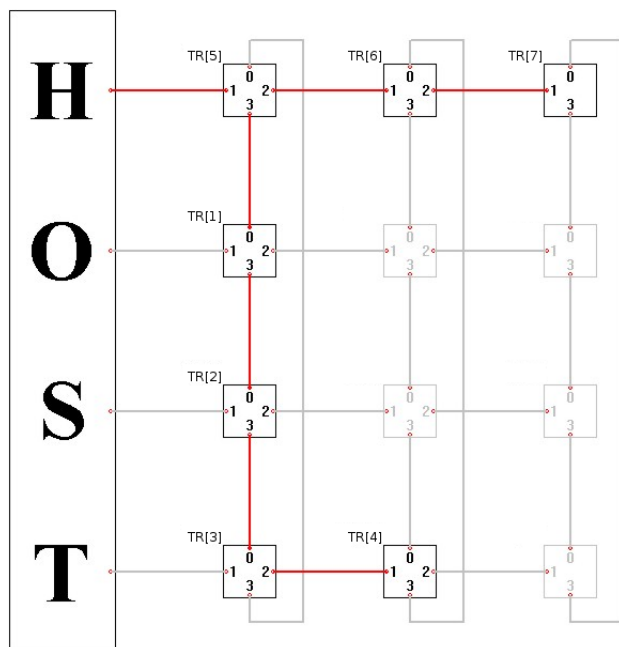
1 Opis zadania

Problem polega na porównywaniu różnych sposobów przetwarzania zadań jednorodnych w systemie transputerowym. Zadana architekturą systemu jest tu łańcuch 7 transputerów z centralnie położonym źródłem danych. Założono, iż w miarę oddalania się od źródła danych, prędkość kolejnego transputera maleje dwukrotnie w porównaniu do poprzedniego. Rozważono następujące sposoby realizacji przetwarzania:

- przetwarzanie szeregowe
- przetwarzanie równoległe z jedną fazą obliczeń i jednym przesłaniem danych - 1f1p
- przetwarzanie równoległe z jedną fazą obliczeń i dwoma przesłaniami danych - 1f2p
- przetwarzanie równoległe z dwoma fazami obliczeń i jednym przesłaniem danych na fazę obliczeń - 2f1p

Ponadto dla każdego z podanych wariantów rozpatrzono dwa sposoby wyboru transputerów, na których realizowane są obliczenia:

- wybór maksymalizujący liczbę szybkich transputerów



Rysunek 1: Model maksymalizujący liczbę szybkich transputerów.

- wybór maksymalizujący liczbę szybkich łącz komunikacyjnych (rysunek!!)

Wszystkie powyższe warianty rozpatrzono teoretycznie, konstruując odpowiednie problemy programowania liniowego, które pozwoliły dobrać optymalną liczbę zadań do przetworzenia dla każdego procesora, która gwarantuje najkrótszy czas przetwarzania całej puli zadań. Aby ułożyć problemy programowania liniowego, przeprowadzono pomiary czasów komunikacji, oraz przetwarzania dla odpowiednich łącz oraz transputerów systemu.

Następnie napisano implementację wariantu z przetwarzaniem 1f1p oraz 1f2p, dla wariantu z maksymalizującą liczbę szybkich łącz oraz dla wariantu maksymalizującego liczbę szybkich procesorów, przydzielając do każdego transputera liczbę zadań wyliczoną z modelu teoretycznego. Wyniki uzyskane z testów powyższych implementacji porównano następnie z modelami teoretycznymi.

2 Wyznaczanie parametrów systemu

Należało wyznaczyć czas inicjalizacji połączenia, prędkość komunikacji, oraz prędkość przetwarzania pojedynczego zadania dla badanego systemu. Uwzględniono fakt, iż czas inicjalizacji połączenia, oraz prędkość przetwarzania pojedynczego zadania, zależy od prędkości transputera - transputery w pierwszej kolumnie od lewej są szybsze niż pozostałe. Natomiast czas komunikacji zależy od rodzaju łącza - łącza poziome są szybsze niż pionowe. Zależności te zostały przedstawione na poniższym rysunku. (rysunek!!!)

2.1 Pomiar czasów inicjalizacji połączenia, oraz prędkości komunikacji

Napisano program mierzący czasy inicjalizacji połączeń, oraz prędkości przesyłania danych. Pomiary dokonano pomiędzy:

- dwoma szybkimi transputerami połączonymi wolnym łączem
- dwoma wolnymi transputerami połączonymi szybkim łączem
- dwoma wolnymi transputerami połączonymi wolnym łączem

Dokonano odpowiedniej synchronizacji procesów, aby upewnić się iż wszystkie one są zainicjowane i gotowe do przeprowadzenia pomiarów. Następnie uruchamiano funkcję ProcTime(), aby określić moment początku komunikacji i wysyłano pakiet danych do drugiego z transputerów, który natychmiast odsyłał otrzymane dane do nadawcy. Nadawca uruchamiał po raz drugi funkcję ProcTime(), aby określić moment końca komunikacji. Zarówno proces nadawcy jak i odbiorcy uruchamiany był z wysokim priorytetem, co gwarantowało wysoką dokładność pomiarów.

Procedurę powtarzano dla różnych wielkości pakietów z danymi. Następnie dla otrzymanych wyników obliczono regresję liniową, dzięki której otrzymano wartości czasu inicjalizacji komunikacji, oraz prędkości komunikacji. Powyższą procedurę powtarzano dla każdej z wymienionych trzech par transputerów. Oto otrzymane wyniki (tabelka z wynikami!!!)

2.2 Pomiar prędkości przetwarzania

Napisano program mierzący prędkość przetwarzania pojedynczego zadania na transputerze szybkim oraz wolnym. Pojedyncze zadanie polegało na obliczeniu metodą Hornera wartości następującego wielomianu: $W(x) = (((4 * x + 12) * x + 3) * x + 3) * x + 7$.

W tym celu na każdym z dwóch transputerów, których dotyczyły pomiary, utworzono po dwa procesy. Proces z priorytetem wolnym, na którym odbywały się obliczenia, oraz proces z priorytetem wysokim, na którym odbywały się pomiary czasu. Wybór takiego sposobu pomiarów wynika z próby jak najdokładniejszego odwzorowania architektury zastosowanej przy implementacji modeli. Po odpowiedniej synchronizacji procesów, która gwarantuje ich gotowość do przeprowadzenia pomiarów, proces obliczeniowy wysyłał sygnał - liczbę całkowitą do procesu pomiarowego i zaczynał obliczenia. Proces pomiarowy po otrzymaniu pierwszego sygnału uruchamiał funkcję ProcTime() do uzyskania czasu rozpoczęcia obliczeń. Proces obliczeniowy po zakończeniu obliczeń wysyłał do procesu mierzącego czas kolejny sygnał. Proces mierzący, po odebraniu drugiego sygnału, uruchamiał funkcję ProcTime() aby uzyskać czas zakończenia obliczeń. Różnica obu czasów dała czas trwania obliczeń powiększony o czas przesyłania komunikatu między procesami (ok. 18 μ s).

Procedurę powtarzano dla różnej liczby zadań do przetworzenia przez procesor, przy czym dla każdej liczby zadań pomiar przeprowadzano dwudziestokrotnie. Następnie dla otrzymanych wyników obliczono regresję liniową, dzięki której otrzymano wartości prędkości obliczeń dla transputera szybkiego oraz wolnego. Oto otrzymane wyniki: (tabelka z wynikami !!!)

3 Opis modeli systemów

Dla każdego wariantu realizacji obliczeń opisanego w punkcie pierwszym, utworzono model matematyczny. Jest nim zadanie programowania liniowego, które następnie rozwiązywano za pomocą programu lp solve. Pozwoliło to uzyskać optymalne rozkłady puli zadań na poszczególne procesory, minimalizując tym samym czas zakończenia obliczeń.

- v - całkowita liczba zadań do realizacji przez system
- T - minimalizowany całkowity czas przetwarzania v zadań

- l_i - liczba zadań dla i tego procesora, $i=1..7$
- a_i - czas przetwarzania pojedynczego zadania na i tym transputerze, $i=1..7$
- s_1, s_2 - czas inicjalizacji komunikacji, odpowiednio dla szybkiego i wolnego transputera
- c_1, c_2 - czas przesłania jednego pakietu danych, odpowiednio wolnym i szybkim łączem

Uwaga!

- Poniżej przedstawiono tylko równania i/lub ograniczenia dla wybranych modeli, bez inicjalizowania zmiennych i stałych problemu. Kompletne kody dla każdego problemu umieszczone zostały na dołączonej płycie CD.
- Przyjęte oznaczenia (indeksy według rysunków z rozdziału Opis zadania)

3.1 Model szeregowy

W modelu tym założono iż komunikacja między transputerami przebiega szeregowo, to znaczy transputer - źródło przesyła dane najpierw do jednego końca łańcucha a następnie do drugiego. Dopiero po zakończeniu przesyłania danych, transputer zaczyna obliczenia. Oto problemy liniowe dla omawianego modelu, przy przyjętych oznaczeniach.(rysunek!!!)

3.1.1 Maksymalizacja liczby szybkich procesorów

minimize obj: $+T$;

$$R1: T \geq s_1 + (l_5 + l_6 + l_7) * c_1 + s_1 + (l_2 + l_3 + l_4) * c_1 + l_1 * a_1;$$

$$R2: T \geq s_1 + (l_5 + l_6 + l_7) * c_1 + s_1 + (l_2 + l_3 + l_4) * c_1 + s_1 + (l_3 + l_4) * c_1 + l_2 * a_2;$$

$$R3: T \geq s_1 + (l_5 + l_6 + l_7) * c_1 + s_1 + (l_2 + l_3 + l_4) * c_1 + s_1 + (l_3 + l_4) * c_1 + s_1 + (l_4) * c_2 + l_3 * a_3;$$

$$R4: T \geq s_1 + (l_5 + l_6 + l_7) * c_1 + s_1 + (l_2 + l_3 + l_4) * c_1 + s_1 + (l_3 + l_4) * c_1 + s_1 + (l_4) * c_2 + l_4 * a_4;$$

$$R5: T \geq s_1 + (l_5 + l_6 + l_7) * c_1 + s_1 + (l_6 + l_7) * c_2 + l_5 * a_5;$$

$$R6: T \geq s_1 + (l_5 + l_6 + l_7) * c_1 + s_1 + (l_6 + l_7) * c_2 + s_2 + (l_7) * c_2 + l_6 * a_6;$$

$$R7: T \geq s_1 + (l_5 + l_6 + l_7) * c_1 + s_1 + (l_6 + l_7) * c_2 + s_2 + (l_7) * c_2 + l_7 * a_7;$$

$$R8: v = l_1 + l_2 + l_3 + l_4 + l_5 + l_6 + l_7;$$

Rozwiązanie problemu programowania liniowego dla tego wariantu:(tabelka!!!)

3.1.2 Maksymalizacja liczby szybkich łączy

minimize obj: $+T$;

$$R1: T \geq s1 + (l5 + l6 + l7) * c1 + s1 + (l2 + l3 + l4) * c1 + l1 * a1;$$

$$R2: T \geq s1 + (l5 + l6 + l7) * c1 + s1 + (l2 + l3 + l4) * c1 + s1 + (l3 + l4) * c2 + l2 * a2;$$

$$R3: T \geq s1 + (l5 + l6 + l7) * c1 + s1 + (l2 + l3 + l4) * c1 + s1 + (l3 + l4) * c2 + s2 + (l4) * c2 + l3 * a3;$$

$$R4: T \geq s1 + (l5 + l6 + l7) * c1 + s1 + (l2 + l3 + l4) * c1 + s1 + (l3 + l4) * c2 + s2 + (l4) * c2 + l4 * a4;$$

$$R5: T \geq s1 + (l5 + l6 + l7) * c1 + s1 + (l6 + l7) * c2 + l5 * a5;$$

$$R6: T \geq s1 + (l5 + l6 + l7) * c1 + s1 + (l6 + l7) * c2 + s2 + (l7) * c2 + l6 * a6;$$

$$R7: T \geq s1 + (l5 + l6 + l7) * c1 + s1 + (l6 + l7) * c2 + s2 + (l7) * c2 + l7 * a7;$$

$$R8: v = l1 + l2 + l3 + l4 + l5 + l6 + l7;$$

Rozwiązanie problemu programowania liniowego dla tego wariantu:(tabelka!!!)

3.2 Model 1f1p

W modelu tym założono, że komunikacja i obliczenia na każdym transputerze przebiega równolegle. Ponadto dla transputera - źródła, komunikacja przebiega równolegle do obu końców łańcucha. Oto problemy liniowe dla omawianego modelu, przy przyjętych oznaczeniach.(rysunek!!!)

3.2.1 Maksymalizacja liczby szybkich procesorów

minimize obj: $+T$;

$$R1: T \geq l1 * a1;$$

$$R2: T \geq s1 + (l2 + l3 + l4) * c1 + l2 * a2;$$

$$R3: T \geq s1 + (l2 + l3 + l4) * c1 + s1 + (l3 + l4) * c1 + l3 * a3;$$

$$R4: T \geq s1 + (l2 + l3 + l4) * c1 + s1 + (l3 + l4) * c1 + s1 + (l4) * c2 + l4 * a4;$$

$$R5: T \geq s1 + (l5 + l6 + l7) * c1 + l5 * a5;$$

$$R6: T \geq s1 + (l5 + l6 + l7) * c1 + s1 + (l5 + l6) * c2 + l6 * a6;$$

$$R7: T \geq s1 + (l5 + l6 + l7) * c1 + s1 + (l5 + l6) * c2 + s2 + (l6) * c2 + l7 * a7;$$

$$R8: v = l1 + l2 + l3 + l4 + l5 + l6 + l7;$$

Rozwiązanie problemu programowania liniowego dla tego wariantu:(tabelka!!!)

3.2.2 Maksymalizacja liczby szybkich łączy

minimize obj: $+T$;

R1: $T \geq l1*a1$;
 R2: $T \geq s1+(l2+l3+l4)*c1 + l2*a2$;
 R3: $T \geq s1+(l2+l3+l4)*c1 + s1+(l3+l4)*c2 + l3*a3$;
 R4: $T \geq s1+(l2+l3+l4)*c1 + s1+(l3+l4)*c2 + s2+(l4)*c2 + l4*a4$;
 R5: $T \geq s1+(l5+l6+l7)*c1 + l5*a5$;
 R6: $T \geq s1+(l5+l6+l7)*c1 + s1+(l5+l6)*c2 + l6*a6$;
 R7: $T \geq s1+(l5+l6+l7)*c1 + s1+(l5+l6)*c2 + s2+(l6)*c2 + l7*a7$;
 R8: $v = l1+l2+l3+l4+l5+l6+l7$;

Rozwiązanie problemu programowania liniowego dla tego wariantu:(tabelka!!!)

3.3 Model 1f2p

W modelu tym założono, że komunikacja odbywa się w dwóch etapach. W pierwszym etapie komunikacji, transputer odbiorca otrzymuje dane potrzebne mu do całkowitej realizacji swojej części obliczeń. W drugim etapie komunikacji otrzymuje natomiast dane do przesłania kolejnym transputerom w łańcuchu. Proces obliczeń i komunikacji został zrównoleglony w maksymalnym możliwym stopniu. Oto problemy liniowe dla omawianego modelu, przy przyjętych oznaczeniach.(rysunek!!!)

3.3.1 Maksymalizacja liczby szybkich procesorów

minimize obj: $+T$;

R1: $T \geq l1*a1$;
 R2: $T \geq s1+(l2)*c1 + l2*a2$;
 R3: $T \geq s1+(l2)*c1 + s1+(l3+l4)*c1 + s1+(l3)*c1 + l3*a3$;
 R4: $T \geq s1+(l2)*c1 + s1+(l3+l4)*c1 + s1+(l3)*c1 + s1+(l4)*c1 + s1+(l4)*c2 + l4*a4$;
 R5: $T \geq s1+(l5)*c1 + l5*a5$;
 R6: $T \geq s1+(l5)*c1 + s1+(l6+l7)*c1 + s1+(l6)*c2 + l6*a6$;
 R7: $T \geq s1+(l5)*c1 + s1+(l6+l7)*c1 + s1+(l6)*c2 + s1*(l7)*c2 + s2*(l7)*c2 + l7*a7$;
 R8: $v = l1+l2+l3+l4+l5+l6+l7$;

Rozwiązanie problemu programowania liniowego dla tego wariantu:(tabelka!!!)

3.3.2 Maksymalizacja liczby szybkich łącz

minimize obj: $+T$;

R1: $T \geq 4 l1*a1$;

R2: $T \geq s1 + (l2) * c1 + l2 * a2$;
 R3: $T \geq s1 + (l2) * c1 + s1 + (l3 + l4) * c1 + s1 + (l3) * c2 + l3 * a3$;
 R4: $T \geq s1 + (l2) * c1 + s1 + (l3 + l4) * c1 + s1 + (l3) * c2 + s1 + (l4) * c2 + s2 + (l4) * c2 + l4 * a4$;
 R5: $T \geq s1 + (l5) * c1 + l5 * a5$;
 R6: $T \geq s1 + (l5) * c1 + s1 + (l6 + l7) * c1 + s1 + (l6) * c2 + l6 * a6$;
 R7: $T \geq s1 + (l5) * c1 + s1 + (l6 + l7) * c1 + s1 + (l6) * c2 + s1 * (l7) * c2 + s2 * (l7) * c2 + l7 * a7$;
 R8: $v = l1 + l2 + l3 + l4 + l5 + l6 + l7$;

Rozwiązanie problemu programowania liniowego dla tego wariantu:(tabelka!!!)

3.4 Model 2f1p

W modelu tym komunikację między transputerami rozbito na dwie fazy. W pierwszej komunikacji fazie każdy z transputerów otrzymuje tylko część potrzebnych mu do obliczeń danych i może wykonać pierwszą fazę obliczeń. Natomiast w drugiej fazie komunikacji każdy z transputerów otrzymuje drugą część potrzebnych mu do obliczeń danych i może realizować drugą fazę obliczeń. Przesyłanie i obliczenia realizowane są w maksymalnie zrównoleżony sposób. Do opisu powyższego modelu niezbędne było wprowadzenie nowych parametrów opisujących system:

- obj - Początek obliczeń na itym transputerze w jotej fazie, $i=1..7$, $j=1..2$, wyjątek dla pierwszego transputera - ob1
- oej - Początek obliczeń na itym transputerze w jotej fazie, $i=1..7$, $j=1..2$, wyjątek dla pierwszego transputera - oe1
- kbij - Początek komunikacji na itym transputerze w jotej fazie, $i=1..6$, $j=1..2$, wyjątek dla pierwszego transputera - kbijk, gdzie k oznacza: g - komunikację w stronę węzła 7, d - w stronę węzła 4
- keij - Początek komunikacji na itym transputerze w jotej fazie, $i=1..6$, $j=1..2$, wyjątek dla pierwszego transputera - keijk, gdzie k oznacza: g - komunikację w stronę węzła 7, d - w stronę węzła 4

Oto problemy liniowe dla omawianego modelu, przy przyjętych oznaczeniach.(rysunek!!!)

3.4.1 Maksymalizacja liczby szybkich procesorów

minimize obj: T;

r1: $T \geq oe1$;

r2: $T \geq oe22$;
 r3: $T \geq oe32$;
 r4: $T \geq oe42$;
 r5: $T \geq oe52$;
 r6: $T \geq oe62$;
 r7: $T \geq oe72$;

r8: $kb21 \geq ke11d$;
 r9: $kb22 \geq ke12d$;
 r10: $kb31 \geq ke21$;
 r11: $kb32 \geq ke22$;
 r12: $kb41 \geq ke31$;
 r13: $kb42 \geq ke32$;
 r14: $kb51 \geq ke11g$;
 r15: $kb52 \geq ke12g$;
 r16: $kb61 \geq ke51$;
 r17: $kb62 \geq ke52$;

r18: $v = l1 + l21 + l22 + l31 + l32 + l41 + l42 + l51 + l52 + l61 + l62 + l71 + l72$;

r19: $ke11g = kb11g + s1 + (l51 + l61 + l71) * c1$;
 r20: $ke12g = kb12g + s1 + (l52 + l62 + l72) * c1$;
 r21: $ke11d = kb11d + s1 + (l21 + l31 + l41) * c1$;
 r22: $ke12d = kb12d + s1 + (l22 + l32 + l42) * c1$;
 r23: $ke21 = kb21 + s1 + (l31 + l41) * c1$;
 r24: $ke22 = kb22 + s1 + (l32 + l42) * c1$;
 r25: $ke31 = kb31 + s1 + (l41) * c2$;
 r26: $ke32 = kb32 + s1 + (l42) * c2$;
 r27: $ke51 = kb51 + s1 + (l61 + l71) * c2$;
 r28: $ke52 = kb52 + s1 + (l62 + l72) * c2$;
 r29: $ke61 = kb61 + s2 + (l71) * c2$;
 r30: $ke62 = kb62 + s2 + (l72) * c2$;

r31: $kb12g \geq ke11g$;
 r32: $kb12d \geq ke11d$;
 r33: $kb22 \geq ke21$;
 r34: $kb32 \geq ke31$;
 r35: $kb42 \geq ke41$;
 r36: $kb52 \geq ke51$;
 r37: $kb62 \geq ke61$;

r38: $ob22 \geq oe21$;
 r39: $ob32 \geq oe31$;

r40: ob42 \geq oe41;
 r41: ob52 \geq oe51;
 r42: ob62 \geq oe61;
 r43: ob72 \geq oe71;

r44: ob21 \geq ke11d;
 r45: ob22 \geq ke12d;
 r46: ob31 \geq ke21;
 r47: ob32 \geq ke22;
 r48: ob41 \geq ke31;
 r49: ob42 \geq ke32;
 r50: ob51 \geq ke11g;
 r51: ob52 \geq ke12g;
 r52: ob61 \geq ke51;
 r53: ob62 \geq ke52;
 r54: ob71 \geq ke61;
 r55: ob72 \geq ke62;

r56: oe1 = ob1 + a1*l1;
 r57: oe21 = ob21 + a2*l21;
 r58: oe22 = ob22 + a2*l22;
 r59: oe31 = ob31 + a3*l31;
 r60: oe32 = ob32 + a3*l32;
 r61: oe41 = ob41 + a4*l41;
 r62: oe42 = ob42 + a4*l42;
 r63: oe51 = ob51 + a5*l51;
 r64: oe52 = ob52 + a5*l52;
 r65: oe61 = ob61 + a6*l61;
 r66: oe62 = ob62 + a6*l62;
 r67: oe71 = ob71 + a7*l71;
 r68: oe72 = ob72 + a7*l72;

Rozwiązanie problemu programowania liniowego dla tego wariantu:(tabelka!!!)

3.4.2 Maksymalizacja liczby szybkich łączy

minimize obj: T;

r1: T \geq oe1;
 r2: T \geq oe22;
 r3: T \geq oe32;
 r4: T \geq oe42;
 r5: T \geq oe52;

r6: $T \geq oe62$;
r7: $T \geq oe72$;

r8: $kb21 \geq ke11d$;
r9: $kb22 \geq ke12d$;
r10: $kb31 \geq ke21$;
r11: $kb32 \geq ke22$;
r12: $kb41 \geq ke31$;
r13: $kb42 \geq ke32$;
r14: $kb51 \geq ke11g$;
r15: $kb52 \geq ke12g$;
r16: $kb61 \geq ke51$;
r17: $kb62 \geq ke52$;

r18: $v = l1 + l21 + l22 + l31 + l32 + l41 + l42 + l51 + l52 + l61 + l62 + l71 + l72$;

r19: $ke11g = kb11g + s1 + (l51 + l61 + l71) * c1$;
r20: $ke12g = kb12g + s1 + (l52 + l62 + l72) * c1$;
r21: $ke11d = kb11d + s1 + (l21 + l31 + l41) * c1$;
r22: $ke12d = kb12d + s1 + (l22 + l32 + l42) * c1$;
r23: $ke21 = kb21 + s1 + (l31 + l41) * c2$;
r24: $ke22 = kb22 + s1 + (l32 + l42) * c2$;
r25: $ke31 = kb31 + s2 + (l41) * c2$;
r26: $ke32 = kb32 + s2 + (l42) * c2$;
r27: $ke51 = kb51 + s1 + (l61 + l71) * c2$;
r28: $ke52 = kb52 + s1 + (l62 + l72) * c2$;
r29: $ke61 = kb61 + s2 + (l71) * c2$;
r30: $ke62 = kb62 + s2 + (l72) * c2$;

r31: $kb12g \geq ke11g$;
r32: $kb12d \geq ke11d$;
r33: $kb22 \geq ke21$;
r34: $kb32 \geq ke31$;
r35: $kb42 \geq ke41$;
r36: $kb52 \geq ke51$;
r37: $kb62 \geq ke61$;

r38: $ob22 \geq oe21$;
r39: $ob32 \geq oe31$;
r40: $ob42 \geq oe41$;
r41: $ob52 \geq oe51$;
r42: $ob62 \geq oe61$;
r43: $ob72 \geq oe71$;

r44: ob21 \geq ke11d;
 r45: ob22 \geq ke12d;
 r46: ob31 \geq ke21;
 r47: ob32 \geq ke22;
 r48: ob41 \geq ke31;
 r49: ob42 \geq ke32;
 r50: ob51 \geq ke11g;
 r51: ob52 \geq ke12g;
 r52: ob61 \geq ke51;
 r53: ob62 \geq ke52;
 r54: ob71 \geq ke61;
 r55: ob72 \geq ke62;

r56: oe1 = ob1 + a1*l1;
 r57: oe21 = ob21 + a2*l21;
 r58: oe22 = ob22 + a2*l22;
 r59: oe31 = ob31 + a3*l31;
 r60: oe32 = ob32 + a3*l32;
 r61: oe41 = ob41 + a4*l41;
 r62: oe42 = ob42 + a4*l42;
 r63: oe51 = ob51 + a5*l51;
 r64: oe52 = ob52 + a5*l52;
 r65: oe61 = ob61 + a6*l61;
 r66: oe62 = ob62 + a6*l62;
 r67: oe71 = ob71 + a7*l71;
 r68: oe72 = ob72 + a7*l72;

Rozwiązanie problemu programowania liniowego dla tego wariantu:(tabelka!!!)

4 Opis implementacji modeli

4.1 Model 1f1p

4.2 Model 2f1p

5 Opis wyników testów implementacji dla wybranych modeli

5.1 Model 1f1p

5.1.1 Maksymalizacja liczby szybkich procesorów

5.1.2 Maksymalizacja liczby szybkich łączy

5.2 Model 1f2p

5.2.1 Maksymalizacja liczby szybkich procesorów

5.2.2 Maksymalizacja liczby szybkich łączy

6 Porównanie wyników implementacji z wynikami uzyskanymi dla modelu teoretycznego

7 Wnioski