Problem:

Jest to problem decyzyjny sprawdzający czy dla podanych danych istnieje rozwiązanie planszy łamigłówki Kakuro

Dane:

E - macierz oznaczająca na których polach będą wyznaczane liczby (e_{ij} = 0), a które są uznane za pełne (czarne pola, pola z sumą: e_{ij} = 1).
E = $[e_{ij}]_{\substack{i=\{1,2,\ldots,m\},\\ j=\{1,2,\ldots,n\}}}$; $e_{ij} \in \{0,1\}$; $m,n \geq 3$; $\forall i,j : e_{i1} = 1, e_{1j} = 1$;

T - macierz oznaczająca położenie pola z sumą, czy ma być to suma w rzędzie ($t_{k3} = 0$) czy w kolumnie ($t_{k3} = 1$) oraz ile wynosi ta suma.
T = $[t_{kl}]_{\substack{k=\{1,2,\ldots,p\},\\ l=\{1,2,3,4\}}}$; $t_{k1} \in i$; $t_{k2} \in j$; $t_{k3} \in \{0,1\}$; $t_{k4} \in \{3,4,\ldots,45\}$;

Dodatkowe dane wyliczone z powyższych danych:

L - Macierz wyznaczająca ile cyfr występuje w danej sumie, jeżeli którakolwiek wartość macierzy L $< 2 \vee > 9$, to znaczy że dane są niepoprawne.
L = $[l_k]_{k=\{1,2,\ldots,p\}}$
$\forall k$:
$\tau := [\tau_q := 0]_{q=\{1,2,\ldots,max(m,n)\}}$
Jeżeli $t_{k3} = 0$:

$$\forall x \in \{t_{k2}+1,\ldots,n\} : \tau_x = \begin{matrix} 1 \leftarrow \sum_{y=t_{k2}+1}^{x} E_{t_{k1}y} = 0 \\ 0 \leftarrow \sum_{y=t_{k2}+1}^{x} E_{t_{k1}y} > 0 \end{matrix}$$

W przeciwnym wypadku ($t_{k3} = 1$):

$$\forall x \in \{t_{k1}+1,\ldots,m\} : \tau_x = \begin{matrix} 1 \leftarrow \sum_{y=t_{k1}+1}^{x} E_{yt_{k2}} = 0 \\ 0 \leftarrow \sum_{y=t_{k1}+1}^{x} E_{yt_{k2}} > 0 \end{matrix}$$

$$l_k = \sum_{x=1}^{max(m,n)} \tau_x$$

Wyznaczyć:

W - Macierz wyników, która dla podanych dobrych danych zawiera rozwiązanie zgodne z ograniczeniami, w przeciwnym wypadku jest zwrócona pusta.
W = $[w_{ij}]_{\substack{i=\{1,2,\ldots,m\},\\ j=\{1,2,\ldots,n\}}}$; $\forall i,j$ jeżeli $e_{ij} = 1$ to $w_{ij} = 0$;

Przy ograniczeniach:

Dla każdej sumy t_k:
- *Cyfry należące do sumy nie mogą się w niej powtórzyć*
- *Cyfry należące do sumy sumują się do niej*

$\forall k$:
Jeżeli $t_{k3} = 0$:

$\forall x \in \{t_{k2}+1,\dots,t_{k2}+l_k\}, y \in \{t_{k2}+1,\dots,t_{k2}+l_k\}, x \neq y : w_{t_{k1}x} \neq w_{t_{k1}y}$;
$\sum_{z=t_{k2}+1}^{t_{k2}+l_k} w_{t_{k1}z} = t_{k4}$
W przeciwnym wypadku ($t_{k3} = 1$) :
$\forall x \in \{t_{k1}+1,\dots,t_{k1}+l_k\}, y \in \{t_{k1}+1,\dots,t_{k1}+l_k\}, x \neq y : w_{xt_{k2}} \neq w_{yt_{k2}}$;
$\sum_{z=t_{k1}+1}^{t_{k1}+l_k} w_{zt_{k2}} = t_{k4}$

Każde pole w macierzy wyników które nie jest równe zero musi $\in \{1,2,3,4,5,6,7,8,9\}$:
$\forall w_{ij} \neq 0 : w_{ij} \in \{1,2,3,4,5,6,7,8,9\}$

Wykorzystując heurystyki:

1. Znając wszystkie możliwe kombinacje sum dla określonej ilości cyfr definiujemy set informacyjny w którym trzymamy jakie cyfry mogą wystąpić w danej kratce. Wykorzystując tą informacje ograniczamy cyfry dla każdej sumy o informacje z setu.

Z - set informacyjny, gdzie $Z_{a,b}$ zwraca listę liczb jakie mogą wystąpić w sumie o danych parametrach a - wartość sumy, b - ile liczb występuje w sumie. Z = $\{z_{a,b}\}_{\substack{a=\{3,4,\dots,45\},\\ b=\{1,2,\dots,9\}}}$

V - kopia macierzy E, w której w miejscu $e_{ij} = 1$ to $v_{ij} = 0$ oraz $e_{ij} = 0$ to v_{ij} jest zestawem liczb od 1 do 9.

V = $[v_{ij}]_{i,j}$; $v_{ij} = \begin{matrix} \{0\} \leftarrow e_{ij} = 1 \\ \{1,2,3,4,5,6,7,8,9\} \leftarrow e_{ij} = 0 \end{matrix}$

$\forall k$:
Jeżeli $t_{k3} = 0$:
$\forall x \in \{t_{k2}+1,\dots,t_{k2}+l_k\}, \forall y \in \{1,2,3,4,5,6,7,8,9\}$:
Jeżeli $y \notin Z_{t_{k4},l_k} \land y \in v_{t_{k1}x}$: Usuń cyfrę y z listy w $v_{t_{k1}x}$
W przeciwnym wypadku ($t_{k3} = 1$) :
$\forall x \in \{t_{k1}+1,\dots,t_{k1}+l_k\}, \forall y \in \{1,2,3,4,5,6,7,8,9\}$:
Jeżeli $y \notin Z_{t_{k4},l_k} \land y \in v_{xt_{k2}}$: Usuń cyfrę y z listy w $v_{xt_{k2}}$

2. Mając ograniczone kombinacje sprawdzamy dla każdej sumy czy występuje jakakolwiek cyfra pojawiająca się tylko raz w kombinacjach danej sumy. Jeżeli taka istnieje to sprawdzamy czy dana wartość sumy dla danej ilości cyfr ma tylko jedną możliwą kombinację cyfr (np. ilość cyfr kombinacji = ilość cyfr). Jeżeli tak jest, to kratkę w której ta cyfra została znaleziona zapisujemy ją, a z kombinacji kratek sumy rzędu i kolumny w której ona występuje usuwamy tą cyfrę oraz ponawiamy próbe sprawdzenia dla zmienionych kombinacji

f := 1
Dopóki f = 1:
f := 0; $\forall k$:
Jeżeli $t_{k3} = 0$:
$\forall y \in \{1,2,3,4,5,6,7,8,9\}$:
$\Phi = [\phi_x]_{x=\{1,\dots,l_k\}}; \phi_x = \begin{matrix} 1 \leftarrow y \in V_{t_{k1},t_{k2}+x} \\ 0 \leftarrow y \notin V_{t_{k1},t_{k2}+x} \end{matrix}$
Jeżeli $\sum_{x=1}^{l_k} \phi_x = 1$:
f := 1; Dla x gdzie $\phi_x = 1$:
Jeżeli $||\phi_x|| \neq 1$:
$V_{z,t_{k2}+x} := \{y\}$
$z \in \{t_{k1}-1,\dots,1\}$; Dopóki $V_{z,t_{k2}+x} \neq \{0\}$: Jeżeli $y \in V_{z,t_{k2}+x} \rightarrow$ usuń cyfre y z listy $V_{z,t_{k2}+x}$
$z \in \{t_{k1}+1,\dots,m\}$; Dopóki $V_{z,t_{k2}+x} \neq \{0\}$: Jeżeli $y \in V_{z,t_{k2}+x} \rightarrow$ usuń cyfre y z listy $V_{z,t_{k2}+x}$

W przeciwnym wypadku ($t_{k3} = 1$) :
$\forall y \in \{1,2,3,4,5,6,7,8,9\}$:
$$\Phi = [\phi_x]_{x=\{1,\ldots,l_k\}}; \phi_x = \begin{matrix} 1 \leftarrow y \in V_{t_{k1}+x,t_{k2}} \\ 0 \leftarrow y \notin V_{t_{k1}+x,t_{k2}} \end{matrix}$$
Jeżeli $\sum_{x=1}^{l_k} \phi_x = 1$:
f := 1; Dla x gdzie $\phi_x = 1$:
Jeżeli $||\phi_x|| \neq 1$:
$V_{t_{k1}+x,z} := \{y\}$
$z \in \{t_{k2}-1,\ldots,1\}$; Dopóki $V_{t_{k1}+x,z} \neq \{0\}$: Jeżeli $y \in V_{t_{k1}+x,z} \rightarrow$ usuń cyfre y z listy $V_{t_{k1}+x,z}$
$z \in \{t_{k2}+1,\ldots,n\}$; Dopóki $V_{t_{k1}+x,z} \neq \{0\}$: Jeżeli $y \in V_{t_{k1}+x,z} \rightarrow$ usuń cyfre y z listy $V_{t_{k1}+x,z}$

3. Otrzymując za pomocą wyżej podanych dwóch heurystyk dla problemu wykorzystujemy metaheurystykę symulowanego wyżarzania, gdzie sąsiad jest wyznaczany poprzez losową zamianę cyfr zawartych w losowej sumie rzędowej. W przypadku dłuższego braku zmian program uruchamia się ponownie od tych samych macierzy dla których się skończył poprzedni SA.

Funkcja wyznaczenia rzędu - fr(częśc macierzy V, suma):

Dla podanych sumy i zawartych do niej pól macierzy V:
Lista wybranych cyfr MC := Pusta lista, pozostała suma sum := suma
Dla każdej listy wyborów cyfr w części macierzy V:
num := los; $los \in$ lista wyborów cyfr;$los \notin MC$; $los \leq sum$
$sum = sum - num$
Wstaw nun do listy MC
Jeżeli sum = 0 oraz ||MC||###### Funkcja wyznaczenia rzędu - fr(częśc macierzy V, suma): = ||częśc macierzy V||: Zwróć MC
W przeciwnym wypadku powtórz funkcję

Funkcja błędu - fe(U - macierz do sprawdzenia):

błąd := 0
$\forall k$:
Jeżeli $t_k 3 = 1$:
błąd = błąd + $|(\sum_{i=t_{k1}+1}^{t_{k1}+l_k+1} U_{i,t_{k2}}) - t_{k4}|$ + 5(Ilość powtarzających się cyfr w $[U_{t_{k1}+1,t_{k2}},\ldots,U_{t_{k1}+l_k+1,t_{k2}}]$) Zwróć błąd

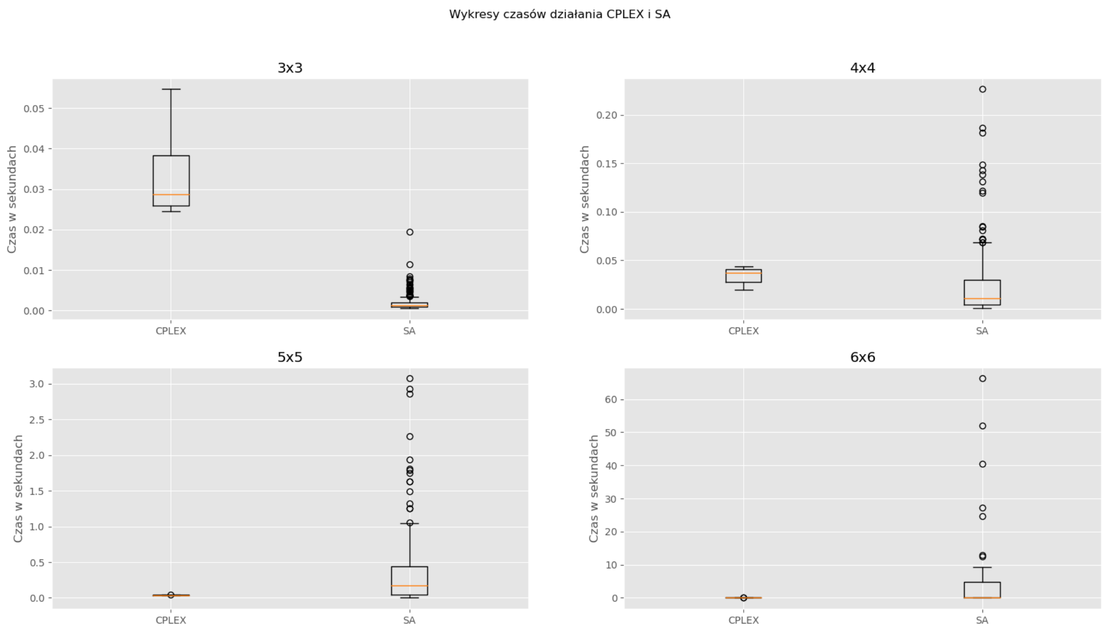
Symulowane wyrzażanie (start_T - temperatura początkowa, max_iter - ile iteracji, alpha - mnożnik wyżarzania):

Wyznaczenie początkowej konfiguracji macierzy U
U := $[0]_{ij}$;
$\forall k$:
Jeżeli $t_k 3 = 0$:
$[U_{t_{k1},t_{k2}+1},\ldots,U_{t_{k1},t_{k2}+l_k+1}]$:= fr($[V_{t_{k1},t_{k2}+1},\ldots,V_{t_{k1},t_{k2}+l_k+1}]$, t_{k4})
błądU := fe(U)

Symulowane wyżarzanie
aktualna_T := start_T
brak_zmian := 0
$\forall \beta \in [1,2,\ldots,max\_iter]$:
$\gamma =$ losowa liczba z przedziału k; $t_{\gamma 3} = 0$

sąsiadU := U
$[sąsiadU_{t_{k1},t_{k2}+1}, \ldots, sąsiadU_{t_{k1},t_{k2}+l_k+1}]$:= fr($[V_{t_{k1},t_{k2}+1}, \ldots, V_{t_{k1},t_{k2}+l_k+1}], t_{k4})$
błądsąsiadU := fe(sąsiadU)
Jeżeli błądsąsiadU = 0: Zwróć sąsiadU jako W
Jeżeli błądsąsiadU > błądU:
U := sąsiadU
błądU := błądsąsiadU
brak_zmian := 0
W przeciwnym wypadku:
Jeżeli liczba losowa rzeczywista z przedziału (0,1) < $\exp(\frac{błądsąsiadU - błądU}{aktualna\_T})$:
U := sąsiadU
błądU := błądsąsiadU
brak_zmian := 0
W przeciwnym wypadku: brak_zmian = brak_zmian + 1
Jeżeli brak_zmian = 10:
Powtórz symulowane wyżarzanie dla aktualnych U i błądU
aktualna_T = aktualna_T * alpha

Wyniki i wnioski:



Powyższy wykres pokazuje, iż jedyne dla najmniejszej możliwej macierzy 3x3 metaheurystyka jest szybsza niż solver CPLEX. Jest to spowodowane tym że konfiguracja problemu przez CPLEX zajmuje dłużej, niż metaheurystyka, natomiast dła większych problemów ewidentnie widać szybsze działanie CPLEX niż metaheurystyk. Do tego od rozmiaru 6x6 zaczynają pojawiać się problemy, iż metaheursytyka nie jest w stanie sobie poradzić z wytworzeniem rozwiązania nawet po 25 iteracjach, natomiast CPLEX nie ma problemu w rozwiązaniu problemów rozmiaru 20x20 w mniej niż 30 sekund.