

Zastosowanie teorii zbiorów przybliżonych w analizie konfliktów

• • • • • • • • • •

Rafał Deja

Praca doktorska

Promotor: prof. dr hab. inż. Andrzej Skowron

Instytut Podstaw Informatyki
Polskiej Akademii Nauk
2000

SPIS TREŚCI

1	Wprowadzenie i motywacja pracy.....	1
1.1	Wstęp.....	1
1.2	Historia analizy konfliktów	2
1.3	Rezultaty pracy	4
1.4	Przykład sytuacji konfliktowej.....	7
2	Model Pawlaka.....	8
2.1	Opis modelu Pawlaka.....	8
2.1.1	Przykład	9
2.2	Analiza sytuacji konfliktowej	10
2.2.1	Przykład	10
2.3	Dlaczego potrzebny jest nowy model?	11
3	Nowy model sytuacji konfliktowych	12
3.1	Stany lokalne	12
3.1.1	Przykład tablicy stanów lokalnych	13
3.1.2	Subiektywna ocena stanu lokalnego (podobieństwo stanów)	14
3.1.3	Przykład subiektywnej oceny	14
3.1.4	Funkcja odległości	15
3.1.5	Lokalne zbiory celów	16
3.1.6	Przykład zbiorów celu, znajdowanie opisów zbiorów z pomocą reguł decyzyjnych	17
3.2	Sytuacje	19
3.3	Ograniczenia (więzy)	20
3.3.1	Przykład ograniczeń	20
3.4	Ocena sytuacji.....	21
3.4.1	Obiektywna ocena sytuacji	22
3.4.2	Preferencje agentów	23
3.5	System z ograniczeniami.....	24
4	Konflikty	25
4.1	Konflikt lokalny.....	25
4.2	Subiektywny konflikt globalny.....	26
4.3	Obiektywny konflikt globalny	27
4.4	Różnorodność konfliktów.....	27
5	Analiza problemu konsensusu	28
5.1	Konsensus na poziomie globalnym (obiektywna ocena)	28
5.2	Konsensus na poziomie globalnym i lokalnym.....	30
5.2.1	Konsensus w bliskim sąsiedztwie celów agentów	31
5.3	Konsensus na poziomie globalnym (ocena agentów).....	34
5.4	Konsensus na poziomie globalnym uwzględniający dobro wspólne i preferencje agentów.....	36
5.5	Algorytmy poszukiwania konsensusu	37
6	Analiza zmienności sytuacji konfliktowych.....	39

6.1	Zasoby.....	39
6.2	Ruchy agentów	40
6.2.1	Koszt zmiany stanu	41
6.2.2	Trwałość sytuacji	42
6.2.3	Funkcja wypłaty, zwycięzca	43
6.3	Pokojowa przestrzeń ruchów	43
6.4	Przestrzeń rozwiązań.....	44
6.4.1	Problem zwycięzcy	45
6.5	Rozwiązania siłowe.....	47
6.5.1	Moc agentów	48
6.5.2	Podział łupów	48
6.5.3	Problem zwycięstwa przy podziale łupów	49
6.5.4	Rozwiązania siłowe a sytuacje trwałe	51
6.5.5	Koalicje	51
7	Analiza strategii.....	54
7.1	Definicja strategii.....	54
7.2	Strategiczna przestrzeń sytuacji.....	56
7.3	Analiza ruchów.....	58
7.4	Przewidywanie konfliktu, zagrożenie konfliktem	59
7.5	Rekonstrukcja sytuacji groźących konfliktem.....	61
8	Strategie obliczeniowe	62
8.1	Wyznaczanie minimalnych reguł decyzyjnych	62
8.2	Obliczanie implikantów pierwszych.....	62
8.2.1	Proste metody redukcji formuł boolowskich	63
8.2.2	Strategia mnożenia formuł	64
8.2.3	Strategia wykluczania (heurystyka)	65
8.2.4	Strategia wykluczania 2 (heurystyka)	68
8.2.5	Strategia wstępnej redukcji	71
8.2.6	Scalanie agentów (clustering)	73
8.3	Programowania z ograniczeniami.....	73
8.4	Dyskretyzacja atrybutów ciągłych.....	73
8.5	Operacje na tablicach decyzyjnych.....	74
9	Program wspomaganie analizy konfliktów	76
9.1	Moduły programu.....	76
9.2	Opis posługiwania się programem	78
9.2.1	Widok projektu	79
9.2.2	Pliki używane w programie	80
9.2.3	Analizy	82
9.2.4	Analiza problemu konsensusu na poziomie globalnym i lokalnym	82
10	Wnioski	85
10.1	Kierunki dalszych badań.....	86
11	Odniesienie do Teorii Gier.....	88
11.1	Podstawowe elementy teorii gier.....	88
11.2	Metagra Howarda.....	91
11.3	Analiza konfliktów (wg Fraser i Hipel [19]).....	94

11.3.1	Porównanie wyników analizy stabilności	95
11.4	Podsumowanie	97
12	Podstawy teorii zbiorów przybliżonych i wnioskowania boolowskiego...	99
12.1	Podstawowe pojęcia	99
12.2	Redukty	101
12.3	Wnioskowanie boolowskie, funkcja rozróżnialności	103
12.4	Minimalne reguły decyzyjne.....	104
12.5	Relacja tolerancji.....	105

SPIS TABEL

Tabela 1 Sytuacja konfliktowa w modelu Pawłaka.....	10
Tabela 2 Tablica rozróżnialności w konflikcie między pracodawcą a pracownikami w modelu Pawłaka.....	11
Tabela 3 Tablica odległości agentów w modelu Pawłaka.....	11
Tabela 4 Stany lokalne agenta ag_1	13
Tabela 5 Stany lokalne agenta ag_1 i ich subiektywna ocena.....	15
Tabela 6 Stany lokalne agenta ag_2 i ich subiektywna ocena.....	15
Tabela 7 Stany lokalne agenta ag_3 z subiektywną oceną.....	15
Tabela 8 Bliskie sąsiedztwo stanu s_2 w stopniu określonym przez parametr $2/3$	16
Tabela 9 Tablica decyzyjna stanów lokalnych agenta ag_1	17
Tabela 10 Tablica rozróżnialności stanów lokalnych agenta ag_1	18
Tabela 11 Przykładowa sytuacja.....	20
Tabela 12 Obiektywna ocena sytuacji (tablica decyzyjna).....	23
Tabela 13 Tablica sytuacji niekonfliktowych.....	31
Tabela 14 Stany lokalne agenta ag_2 z bliskim sąsiedztwem.....	33
Tabela 15 Stany lokalne agenta ag_3 z bliskim sąsiedztwem.....	33
Tabela 16 Moc agentów w konflikcie między pracodawcą i pracownikami....	48
Tabela 17 Subiektywna ocena różnych sytuacji.....	52
Tabela 18 Tablica odległości - zbieżności poglądów agentów.....	52
Tabela 19 Tablica przejścia (dane historyczne).....	55
Tabela 20 Przykładowa sytuacja.....	57
Tabela 21 Porównanie podstawowych elementów modelu zdefiniowanego w pracy i modelu opartego o teorię gier.....	89
Tabela 22 Postać normalna konfliktu między US i USSR.....	90
Tabela 23 Metagra – konflikt między US a USSR.....	92
Tabela 24 Wyniki analizy metagry w konflikcie między US a USSR.....	93
Tabela 25 Tablica analizy stabilności konfliktu między U.S. i USSR.....	95
Tabela 26 Wyniki trzech typów analizy stabilności konfliktu między US a USSR.....	97

SPIS ILUSTRACJI

Rysunek 1 Schemat możliwości wykonania ruchu dla układu dwóch agentów biorących udział w konflikcie.	41
Rysunek 2 Program analizy konfliktów (Conflict Analysis Center) widok po otwarciu projektu.	78
Rysunek 3 Menu kontekstowe konfliktu.	79
Rysunek 4 Dodawanie agentów do projektu konfliktu.	80
Rysunek 5 Przykłady plików używanych w programie analizy konfliktów.	81
Rysunek 6 Wynik analizy problemu konsensusu na poziomie globalnym i lokalnym.	84

PODZIĘKOWANIA

Autor pragnie serdecznie podziękować promotorowi profesorowi Andrzejowi Skowronowi za inspirację do podjęcia tej tematyki, wszelki trud i okazaną pomoc. Profesor Skowron, ze stanowczym spokojem, zachęcał mnie do podejmowania kolejnych wysiłków przekazując swoje uwagi i cenne pomysły. Dziękuję za okazaną mi życzliwość.

Mojej rodzinie za wyrozumiałość, cierpliwość i pomoc w trudnych chwilach.

WPROWADZENIE I MOTYWACJA PRACY

Głównym celem pracy jest opracowanie metody modelowania rzeczywistych sytuacji konfliktowych, zapewniającej ich wielostronną analizę. Inspiracją i motywacją do pojęcia tej pracy jest model profesora Pawłaka oparty na teorii zbiorów przybliżonych [38], [29]. Jego niezwykłą zaletą jest to, że umożliwia prosty opis skomplikowanych, rzeczywistych sytuacji konfliktowych. Rozszerzając podejście Pawłaka autor zaproponował model, który pozwala analizować przyczyny konfliktu i daje podstawę do poszukiwania konsensusu. Siłę modelu stanowi możliwość oparcia analizy na rzeczywistych danych (danych eksperymentalnych), informacjach pochodzących od stron konfliktu. Co więcej opracowane metody analizy pozwalają na wnioskowanie przy niepewnej i niepełnej informacji. Zastosowanie teorii zbiorów przybliżonych i wnioskowania boolowskiego pozwoliło stworzyć nowy model sytuacji konfliktowych i opracować metody analizy i rozwiązywania konfliktów. W oparciu o te metody powstał komputerowy system wspomagania decyzji: program analizy konfliktów.

1.1 Wstęp

Wraz z upowszechnieniem się rozproszonych systemów komputerowych rośnie znaczenie modeli interakcji i kooperacji agentów (ang. multi-agent systems). Przez interakcje rozumiemy działania grupy agentów, które mają prowadzić do osiągnięcia celów agentów, w sytuacji ograniczonej dostępnymi zasobami [52], [57]. Jeżeli ilość zasobów jest niewystarczająca do bezpośredniego osiągnięcia celów agentów dochodzi do konfliktów między agentami. Problem kooperacji polega natomiast na uzgodnieniu takiego podziału zasobów między agentami, aby osiągnąć wspólny cel (lub poszczególne cele agentów) – jest rozwiązaniem konfliktu przez współdziałanie. Komputerowo wspomagana analiza konfliktów jest niezbędna, gdy inteligentni agenci (komputery, programy, roboty) wchodzić ze sobą w interakcje. Dzieje się tak przede wszystkim za sprawą gwałtownego wzrostu wykorzystania światowej sieci komputerowej (Internetu). Automatyzacja handlu, obsługa kontrahentów, negocjacje w sprawie ceny, daty i miejsca spotkania, kooperacje (wspólne

przedsięwzięcia) firm często bardzo odległych od siebie, uzgadnianie wspólnych operacji wojskowych itd. są przykładami procesów, których automatyzacja prowadzi do konieczności opracowania komputerowo wspomaganą analizy konfliktów. Informatyka jest przykładem jednej z wielu dziedzin, w których pojawiają się konflikty. A pojawiają się one niemal w każdej dziedzinie ludzkiego życia: w polityce, bussinesie, sporcie, zarządzaniu, w stosunkach międzypaństwowych i międzyludzkich, podczas prowadzenia operacji wojskowych czy rozpraw sądowych itd., słowem konflikty nieodłącznie towarzyszą człowiekowi. Dlatego też człowiek od dawna stara się dociekać przyczyn konfliktów, racjonalizuje je i rozwiązuje. Znaczenie analizy konfliktów podkreśla fakt powstania zawodu negocjatora czy arbitra. Na większych uczelniach istnieją ośrodki zajmujące się analizą konfliktów (Conflict Analysis Center), a w Internecie proponują swoje usługi zawodowi negocjatorzy. Większość konfliktów jest rozwiązywana na drodze pokojowej (poprzez uzgadnianie stanowisk, głosowanie czy negocjacje), choć czasem rozwiązania siłowe wydają się stronom korzystniejsze.

Wyzwaniem jest opracowanie metody modelowania konfliktów. Niezależnie od skali konfliktów czy też spraw, których dotyczą chcielibyśmy modelować konflikty w ten sam sposób – zawrzeć w takich samych strukturach matematycznych i logicznych. Niezależnie więc od tego, czy mamy do czynienia z międzynarodowym konfliktem zbrojnym czy biorą w nim udział strony o światowym znaczeniu, czy też konflikt jest lokalny modelując sytuację konfliktową należy wskazać przyczyny konfliktu i szukać rozwiązania (konsensusu). W zaproponowanym w pracy modelu wyniki nie zależą od typu, wielkości czy znaczenia konfliktu, ale od jakości danych opisujących sytuację. Oczywiście, przy bardzo złożonych problemach powstają trudności związane ze zgromadzeniem odpowiednich danych jak i wymagany czas analizy może być znaczny.

1.2 Historia analizy konfliktów

Historycznie, najwcześniej próby analizy konfliktów były prowadzone w oparciu o teorię gier. Klasyczna teoria gier nie daje jednak podstaw dla uniwersalnych rozwiązań przy grach o sumie niezerowej, czy dla wielu agentów (zwłaszcza gdy nie jest znany dokładny model). Wiele zadowalających wyników otrzymano np. opierając się na *metagrze Howarda* [24], [19]. W tym podejściu bierze się pod uwagę reakcje danego gracza na znane strategie

pozostałych graczy. Pełniejsze porównanie modelu opartego o teorię gier i zbiory przybliżone zostało zamieszczone w rozdziale 11.

Powstało też wiele modeli analitycznych, opracowanych dla wielu klas konfliktów i w różnym zakresie szczegółowości. Istotna z nich część odnosi się do konfliktów zbrojnych (modele walki). Do najbardziej znanych należą modele walki Lanchestera [49], które opierają się na równaniach różniczkowych. Równania różniczkowe są też wykorzystane w grach różniczkowych, których najbardziej znanym zastosowaniem jest modelowanie walki powietrznej [33]. W innych modelach analitycznych najczęściej stosuje się metody statystyczne do analizy konfliktów. Mając określone reakcje agenta na decyzje przeciwników można zbudować proces Markowa rozwoju konfliktu [48]. Ponadto metody statystyczne często wprowadza się do modeli opartych o teorię gier (na przykład gry w warunkach niepewności – patrz 11.1).

Problem poszukiwania konsensusu występuje też często w zagadnieniach decyzyjnych. Szczególnym przypadkiem zagadnień decyzyjnych jest ogólny problem wyboru [31], [1], [3]. Opracowany model wyboru (ang. consensus methods) polega na doborze odpowiedniej funkcji konsensusu w zależności od zadanego problemu decyzyjnego [11], [16], [30]. Najbardziej elementarny konflikt w systemach decyzyjnych występuje wtedy, gdy za pomocą dostępnych danych obiekt może zostać zaklasyfikowany do więcej niż jednej klasy decyzyjnej (reguły wskazują na różne wartości decyzji). Strategie rozstrzygania tego typu konfliktów opierają się na dodatkowych informacjach, takich jak: postać reguł decyzyjnych, znaczenie poszczególnych reguł, poziom zaufania, postać klas decyzyjnych, cel klasyfikacji itd. Najbardziej znane metody rozstrzygania konfliktów między regułami decyzyjnymi zostały omówione w punkcie 8.5.

W modelu negocjacji [26], [27] zastosowanym w systemie Negoplan zaproponowano szkieletowy system ekspertowy do symulacji negocjacji między agentami. W modelu problem jest reprezentowany przez hierarchiczną strukturę reguł decyzyjnych (goal decomposition model). Sprawy, których negocjacje dotyczą, są przedstawione w postaci faktów. Tworzą one znajdujące się na dole struktury hierarchicznej, odpowiednie reguły decyzyjne. Odpowiedź

danego agenta na nową pozycję oponentów jest wnioskowana w oparciu o strukturę reguł decyzyjnych.

Jak wspomniano we wstępie (punkt 1.1) rozwój rozproszonych systemów komputerowych powoduje wzrost znaczenia systemów wieloagentowych, w których "inteligentni" agenci mają ze sobą współpracować, kooperować. Zadaniem każdego agenta jest wykonanie swojego zadania często przy współpracy z innymi agentami. Zadania agentów mogą być jednak w różny sposób sprzeczne (sprzeczne cele, plany czy założenia [5]) i ograniczone przez dostępne zasoby. Szczególne znaczenie ma więc w tych systemach komputerowo wspomagane rozwiązywanie konfliktów. Różne aspekty analizy konfliktów w systemach wieloagentowych są przedmiotem prac zespołu pod kierownictwem prof. Tuomasa Sandholma na Uniwersytecie Washingtona w St. Louis (Multiagent System Research Group). Głównie w oparciu o teorię gier prowadzone są badania na temat formowania koalicji, negocjacji między agentami a także automatycznego handlu w sieci czy obsługi kontrahentów – problemy alokacji zasobów i podziału zadań między agentami o różnym stopniu autonomii [53], [54], [55], [56], [57], [64]. Metodami automatyzacji negocjacji w oparciu o teorię gier, ale także klasyfikacją i rozpoznawaniem konfliktów zajmowali się również inni autorzy [16], [51], [52], [72], [73]. W systemach wieloagentowych, w których agenci obdarzeni są pewną autonomią trudno sobie wyobrazić zbudowanie modelu analitycznego zachowania agentów z uwagi na liczbę, różnorodność i złożoną strukturę agentów oraz niepełną informację o ich zachowaniu. Dlatego też w pracy proponuje się modelowanie zachowania agentów w oparciu o dane eksperymentalne, o ich zachowanie w przeszłości.

1.3 Rezultaty pracy

W pracy proponuje się zastosowanie zbiorów przybliżonych [41], [29], [45], [46] i wnioskowania boolowskiego [9] do modelowania sytuacji konfliktowych, analizy i poszukiwania konsensusu. Zastosowany model jest rozwinięciem modelu zaproponowanego przez Pawlaka [39], [42], [43]. Podstawą nowego modelu są informacje zebrane od każdej ze stron biorących udział w konflikcie. Model jest oparty na danych eksperymentalnych a nie analitycznych – (prócz ograniczeń, którą mogą być zadane w różny sposób) nie wprowadza się bezpośrednio żadnych analitycznych zależności funkcyjnych do opisu konfliktu.

Analizując stany lokalne, preferencje agentów i sytuacje globalne poznajemy najbardziej konfliktowe sprawy. Ograniczenia zaś są fundamentalną przyczyną konfliktów. Często uzgodnione rozwiązanie jest niekorzystne dla „słabych” agentów, dlatego w modelu jest możliwość uwzględnienia „dobra wspólnego” wyspecyfikowanego przez arbitra w danym konflikcie.

Badania sytuacji konfliktowych są początkowo prowadzone w stanie ustalonym, gdy działania agentów są zawieszane, na przykład, na czas negocjacji. Ciekawym problemem, szeroko omawianym w pracy, jest również analiza ruchów agentów, czyli dynamicznych zmian stanów lokalnych zgodnie ze strategią danego agenta i/lub w odpowiedzi na ruchy pozostałych agentów biorących udział w konflikcie. Na podstawie strategii agentów i oceny sytuacji globalnej można analizować zachowanie się układu agentów i monitorować sytuacje powstające w tym układzie. Staje się możliwe ostrzeżenie przed zbliżającym się konfliktem i wskazywanie strategii rekonstrukcji sytuacji do sytuacji *bezpiecznej*.

Teoria zbiorów przybliżonych [41], [29] zastosowana w pracy, służy w pierwszym etapie do gromadzenia i przetwarzania danych eksperymentalnych – rzeczywistych danych zebranych od stron konfliktu. Duża liczba danych, ich niespójność i niekompletność często sprawia wiele problemów. W oparciu o teorię zbiorów przybliżonych wprowadzono wiele metod redukcji nadmiarowej informacji zarówno obiektów (przykładów) jak i atrybutów. (Więcej szczegółów na ten temat można znaleźć w punkcie 8.5). Przykładowo w pracy redukcja atrybutów jest wykonywana poprzez wyznaczenie reguł decyzyjnych w oparciu o minimalne redukty (np. w punkcie 3.1.6). Często stosuje się też dyskretyzację atrybutów ciągłych czy wprowadza się atrybuty uogólnione (punkt 8.4). Na przykład używany w przykładzie atrybut „zysk zakładu” został uogólniony tak, że jego wartości oznaczają poziom niski, średni i wysoki a nie konkretną wartość liczbową (często trudną do zmierzenia). Warto również zwrócić uwagę na przejrzystość opisu sytuacji konfliktowych w oparciu o tablice informacyjne. Najważniejszym jednak zastosowaniem teorii zbiorów przybliżonych w pracy jest wnioskowanie przy niepełnej informacji – klasyfikacja nowych przypadków. W sytuacjach konfliktowych w połączeniu z wnioskowaniem aproksymacyjnym pozwala na szukanie rozwiązań wśród sytuacji nie rozważanych przez agentów. Ma to ogromne znaczenie praktyczne. Rzeczywiste konflikty są przeważnie pozbawione precyzyjnego,

pełnego opisu, a rozwiązanie jest zazwyczaj następstwem ustępstw każdej ze stron. Punktem wyjścia do rozwiązania każdego problemu przedstawionego w pracy jest generacja reguł decyzyjnych na podstawie danych zawartych w tablicy informacyjnej i wprowadzanego atrybutu decyzyjnego. W pracy zastosowano generację reguł decyzyjnych w oparciu o minimalne redukt, ponieważ najlepiej klasyfikują one nowe przypadki. Dalej stosuje się wnioskowanie boolowskie [9], [44], [41] do rozwiązywania problemów. Metoda ta polega ogólnie na zakodowaniu problemu optymalizacyjnego przez odpowiadającą formułę boolowską w taki sposób, że każdy implikant pierwszy tej formuły wyznacza rozwiązanie problemu. Wnioskowanie boolowskie okazało się bardzo dobrym narzędziem do definiowania i rozwiązywania problemów dotyczących analizy konfliktów.

W pracy zaproponowano nowy model sytuacji konfliktowej i przedstawiono w nim wiele ważnych analiz. Dzięki zdefiniowaniu pojęcia konfliktu jest możliwe poszukiwanie rozwiązań – konsensusu. Problemowi szukania konsensusu poświęcono szczególnie dużo uwagi (rozdział 5). Zaproponowano algorytmy rozwiązywania konfliktów globalnych i lokalnych z uwzględnieniem dobra wspólnego i preferencji agentów. Sposoby działania algorytmów i wyniki są zaprezentowane na przykładzie konfliktu między pracodawcą i pracownikami reprezentowanymi przez dwa związki zawodowe. W dalszej części pracy (rozdział 6) rozważa się zmienność sytuacji konfliktowych spowodowaną ruchami agentów. Wyznaczenie przestrzeni ruchów agentów w sytuacji naturalnie ograniczonej zasobami i stanami innych agentów jest najbardziej podstawowym problemem rozważanym w tym rozdziale. Szczególne znaczenie ma jednak analiza problemu "zwycięzcy" (punkty 6.4.1 oraz 6.5.3), zwłaszcza w sytuacji gdy agenci decydują się na rozwiązania siłowe. Natomiast na podstawie analizy zebranych danych historycznych o ruchach agentów, rozważamy problem przewidywania konfliktu i możliwość rekonstrukcji sytuacji grożących konfliktem (rozdział 7). Prócz przedstawionych analiz wiele innych problemów związanych z analizą konfliktów zostało zasygnalizowanych w pracy i zaproponowano ich rozwiązania. Należą tu zwłaszcza problemy związane z dynamiką sytuacji konfliktowych jak np. analizy w grafie zmienności, poszukiwanie koalicji czy analiza kosztów związanych ze zmianą stanu danego agenta. Zaproponowany model można również zastosować w rozwiązywaniu problemów decyzyjnych w systemach wieloagentowych (zadania wymagające kooperacji, uzgadniania celów agentów, alokacja zasobów).

Ważnym rezultatem pracy jest opracowanie komputerowego programu wspomaganie decyzji w sytuacjach konfliktowych. Program jest przystosowany do analizy i proponowania rozwiązań rzeczywistych sytuacji konfliktowych (wszystkie wyniki przedstawione w pracy zostały otrzymane przy pomocy programu). Program został zaimplementowany w taki sposób, że umożliwia wszechstronną analizę konfliktów (nie ogranicza się do analiz wprowadzonych w pracy). Budowa programu opiera się na ogólnych modułach realizujących problemy znane z teorii zbiorów przybliżonych: jak wyznaczanie minimalnych reguł decyzyjnych i redukcji złożonej formuły boolowskiej oraz na module wyznaczającym formuły boolowskie na podstawie zależności przedstawiających ograniczenia. Szczególną uwagę autor poświęcił implementacji modułu redukcji formuł boolowskich tak by realizował on strategię doboru formuł do redukcji i był optymalny ze względu na zajętość pamięci i czas obliczeń. Dodatkowo program realizuje interfejs przyjazny dla użytkownika.

Podsumowując, najważniejszymi rezultatami pracy są:

- opracowanie modelu sytuacji konfliktowych,
- zaproponowanie metod analizy konfliktów i algorytmów rozwiązywania problemów w tym modelu,
- opracowanie komputerowego systemu wspomaganie analizy konfliktów.

1.4 Przykład sytuacji konfliktowej

Rozważmy przykład sytuacji konfliktowej, która mogła powstać w zakładzie pracy. Przykład oparto na rzeczywistym konflikcie zaobserwowanym przez autora, który zmodyfikowano i uproszczono tak aby służył do czytelnej ilustracji pojęć prezentowanych w całej pracy.

Konflikt powstał między pracodawcą i pracownikami reprezentowanymi przez dwa związki zawodowe TU1 i TU2. Pracodawca jest zainteresowany przede wszystkim w jak największym zysku zakładu, w nowych inwestycjach i może satysfakcji pracowników z zakładu i swojej pracy. Dla związku zawodowego TU1 najważniejsze są poziom wynagrodzenia pracowników i dobre warunki pracy, a dla TU2 płaca, opieka socjalna ale i poziom zatrudnienia. W dalszej części pracy dokładniej zdefiniujemy tę sytuację konfliktową i poddamy ją analizie.

MODEL PAWLAKA

Okolo roku 1984 Pawlak zaproponował model sytuacji konfliktowych oparty o systemy informacyjne i teorię zbiorów przybliżonych [39]. Od tego czasu jest on rozwijany przez różnych autorów (np. [43], [12], [13]), głównie poprzez wprowadzanie nowych analiz. Model ten opisuje sytuację konfliktową w stanie ustalonym, w którym agenci zdecydowali się analizować konflikt metodą pokojową poprzez głosowanie na zadane konfliktowe sprawy. W tak opisaney sytuacji w prosty sposób wnioskujemy o istniejących koalicjach czy też obliczamy natężenie konfliktu. Niezależnie od wspomnianego opisu sytuacji konfliktowej Pawlak rozważa rozwiązania konfliktu na zasadzie podziału łupu (siłowe i przy wspólnej zgodzie) [40]. W rozdziale tym dokładnie zdefiniujemy model Pawlaka.

2.1 Opis modelu Pawlaka

W konflikcie biorą udział strony nazywane *agentami*. Agenci definiują swoje stanowisko przypisując opinię każdej zadanej (wybranej) sprawie. W modelu Pawlaka zależności pomiędzy agentem a daną sprawą (atrybutem) są przedstawiona w postaci tablicy informacyjnej gdzie agentom odpowiadają wiersze tablicy a sprawom, atrybuty w kolumnach. Wartości przyporządkowane dla każdego agenta i atrybutu pochodzą z trójwartościowego zbioru $\{-1, 0, 1\}$, gdzie -1 oznacza, że agent jest *przecim*, 0 jest *neutralny* i 1 oznacza *aprobatę* dla danej sprawy. Formalnie taka tablica tworzy *system informacyjny* zdefiniowany w następujący sposób.

System informacyjny modelujący konflikty jest parą $I=(U, A)$, gdzie:

U – jest niepustym, skończonym zbiorem nazywanym *universum*; elementami zbioru są obiekty – tutaj agenci,

A – jest niepustym, skończonym zbiorem atrybutów (spraw).

Każdy atrybut $a \in A$ jest odwzorowaniem: $a: U \rightarrow V_a$, gdzie zbiór V_a jest zbiorem wartości atrybutu a . Elementy zbioru V_a są opiniami agenta ag o sprawie a . Zbiory wartości

atrybutów dla modelu konfliktów Pawlaka są ograniczone do trzech wartości: $\{-1, 0, 1\}$, które oznaczają odpowiednio *sprzecim*, *neutralność*, *aprobatę*.

Relacje między agentami są zdefiniowane poprzez ich poglądy na rozważane sprawy – co wyraża funkcja $\phi_a(ag, ag')$ określona następująco:

$$\phi_a(ag, ag') = \begin{cases} 1 & \text{gdy } a(ag)a(ag')=1 \quad \text{lub } ag = ag' \\ 0 & \text{gdy } a(ag)a(ag')=0 \quad \text{i } ag \neq ag' \\ -1 & \text{gdy } a(ag)a(ag')=-1 \end{cases}$$

Agenci ag, ag' są w relacji *zgody* na atrybucie a w zbiorze U^2 , co zapisujemy $R_a^+(ag, ag')$ wtedy i tylko wtedy gdy $\phi_a(ag, ag') = 1$, relacji *neutralności* $R_a^0(ag, ag')$ wtedy i tylko wtedy gdy $\phi_a(ag, ag') = 0$ i *konflikcie* $R_a^-(ag, ag')$ wtedy i tylko wtedy gdy $\phi_a(ag, ag') = -1$. Relacja zgody jest relacją *równoważności* [42], agenci należący do klasy równoważności wyznaczonej przez tę relację są w *koalicji* określonej przez atrybut a .

2.1.1 Przykład

Jak wspomniano w modelu Pawlaka agenci wyrażają opinie na wybrane sprawy konfliktowe. Niech dla naszego przykładu, konfliktu między pracodawcą a pracownikami będą dane następujące sprawy:

a – wzrost wynagrodzenia pracowników,

b – poprawa warunków pracy,

c – zwiększenie opieki socjalnej, gwarancja zatrudnienia,

d – zwiększenie zysku zakładu poprzez redukcję zatrudnienia i zmianę systemu wynagrodzeń,

e – zwiększenie poziomu inwestycji w następstwie zwiększenia zysku.

Model dla rozważanego konfliktu jest opisany tablicą informacyjną [37], gdzie ag_1, ag_2, ag_3 oznaczają odpowiednio *pracodawcę*, i związki *TU1* i *TU2*:

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>
<i>ag₁</i>	-1	0	-1	1	1
<i>ag₂</i>	1	1	0	0	-1
<i>ag₃</i>	0	1	1	-1	-1

Tabela 1 Sytuacja konfliktowa w modelu Pawlaka

Agenci *ag₂* i *ag₃* są w koalicji w sprawach *b* i *e*.

2.2 Analiza sytuacji konfliktowej

Różnice w opiniach agentów najlepiej analizować przekształcając tablicę informacyjną do postaci *tablicy rozróżnialności* [29], [44]. Dalej różnice te można wyrazić w sposób liczbowy stosując *funkcję odległości* zdefiniowaną na przykład w następujący sposób:

$$\rho_B^*(ag, ag') = \frac{\sum_{a \in B} \phi_a^*(ag, ag')}{|B|}, \text{ gdzie}$$

$$\phi_a^*(ag, ag') = \frac{1 - \phi_a(ag, ag')}{2} = \begin{cases} 0 & \text{gdy } a(ag)a(ag') = 1 \quad \text{lub } ag = ag' \\ 0.5 & \text{gdy } a(ag)a(ag') = 0 \quad \text{i } ag \neq ag' \\ 1 & \text{gdy } a(ag)a(ag') = -1 \end{cases}$$

Aby rozwiązać sytuację konfliktową agenci mogą negocjować konsensus albo wzajemnie się zwalczać. Próby wprowadzenia negocjacji do modelu oraz propozycje szukania konsensusu były prowadzone między innymi przez autora w pracach [12], [13]. Dla rozwiązań siłowych można też zdefiniować *strategię odstraszenia* [40], która jeśli istnieje mówi o tym czy dany agent ma wystarczającą *siłę* aby odstraszyć swoich wrogów. Szczegółowe analizy w modelu Pawlaka można znaleźć między innymi w pracach [42], [43], [12], [13].

2.2.1 Przykład

Tabela 2 i Tabela 3 przedstawiają odpowiednio macierz *rozróżnialności* i wartości funkcji odległości dla konfliktu między pracodawcą a pracownikami.

	ag_1	ag_2	ag_3
ag_1			
ag_2	a b c d e		
ag_3	a b c d e	a c d	

Tabela 2 Tablica rozróżnialności w konflikcie między pracodawcą a pracownikami w modelu Pawłaka

	ag_1	ag_2	ag_3
ag_1			
ag_2	0.7		
ag_3	0.8	0.3	

Tabela 3 Tablica odległości agentów w modelu Pawłaka

2.3 Dlaczego potrzebny jest nowy model?

Przedstawiony model sytuacji konfliktowej pozwala analizować podstawowe własności konfliktów. W formalny sposób opisuje relacje między stronami biorącymi udział w konflikcie i wprowadza metody analizy i wnioskowania w takich sytuacjach.

Jednak sytuacja konfliktowa jest w tym modelu opisywana na bardzo zewnętrznym poziomie. Konflikt jest definiowany przez zestawienia opinii agentów (głosowanie) na sprawy konfliktowe (*sprzeciw*, *neutralność*, *aprobatę*). Jak można zauważyć nawet na prostym przykładzie konfliktu między pracodawcą i pracownikami, wybranie (zdefiniowanie) spraw konfliktowych jest bardzo trudne. Rzadko też w rzeczywistych warunkach można sprowadzić ocenę danej sprawy do jednej z trzech wartości $\{-1, 0, 1\}$. Przede wszystkim jednak nie docieka się w tym modelu prawdziwej przyczyny konfliktu. Wszak opinie agentów na dane sprawy są tylko ich decyzjami podjętymi na podstawie lokalnej wiedzy, oceny stanu posiadania czy realizowanej strategii. Nieobecność przyczyn konfliktu w tym modelu uniemożliwia poszukiwanie rozwiązania – proponowanie konsensusu. Powyższe przyczyny skłaniają do wprowadzenia nowego modelu, opartego jednak na tych samych metodach formalnych.

NOWY MODEL SYTUACJI KONFLIKTOWYCH

Nowy model sytuacji konfliktowych jest rozszerzeniem modelu Pawlaka opisanego w poprzednim rozdziale. Głównym celem wprowadzenia nowego modelu jest umożliwienie zdefiniowania i analizy, w oparciu o ten model, przyczyn konfliktu oraz poszukiwanie rozwiązania – konsensusu. Uwzględniono więc lokalne opisy sytuacji konfliktowej, preferencje agentów i ograniczenia, więzy, które uzależniają agentów wzajemnie od siebie. Ograniczenia wynikają głównie ze skończoności różnego typu zasobów i są fundamentalną przyczyną konfliktów (choć bezpośrednim powodem powstawania konfliktów jest oczywiście chęć posiadania zasobów). Dalej wprowadziliśmy ocenę sytuacji globalnych, która może być obiektywna (dana przez arbitra) lub wynikać z preferencji agentów. W rozdziale tym dokładnie zdefiniujemy wszystkie te pojęcia, budując nowy model sytuacji konfliktowych.

3.1 Stany lokalne

Niech będzie dany zbiór agentów Ag biorących udział w konflikcie. Informacja o zbiorze stanów lokalnych U_{ag} agenta $ag \in Ag$ może być reprezentowana (podobnie jak w modelu Pawlaka) w formie tablicy informacyjnej. Formalnie z każdym agentem ag wiążemy systemem informacyjny $I_{ag} = (U_{ag}, A_{ag})$, gdzie $a: U_{ag} \rightarrow V_a$ dla każdego $a \in A_{ag}$, a V_a jest zbiorem wartości atrybutu a . Przyjmujemy oznaczenie:

$$V_{ag} = \bigcup_{a \in A_{ag}} V_a$$

Każdy stan lokalny $s \in U_{ag}$ jest dokładnie opisany przez informację o nim $Inf_{A_{ag}}(s)$, gdzie $Inf_{A_{ag}}(s) = \{(a, a(s)): a \in A_{ag}\}$. Zbiór $\{Inf_{A_{ag}}(s): s \in U_{ag}\}$ jest oznaczany przez $INF_{A_{ag}}$ i nazywany *zbiorem informacji* agenta ag . Zakładając, że $A_{ag} = \{a_1, \dots, a_n\}$ możemy informację $Inf_{A_{ag}}(s)$ o stanie lokalnym s identyfikować z wektorem informacji $(a_1(s), \dots, a_n(s))$.

Zakładamy, że zbiory $\{A_{ag}\}$ są parami rozłączne to znaczy $A_{ag} \cap A_{ag'} = \emptyset$ dla $ag \neq ag'$. Warunek ten oznacza, że każdy agent opisuje sytuację po swojemu. Sposób pojmowania „tego samego świata” przez różnych agentów może być zupełnie inny. Zależności między atrybutami różnych agentów będą zdefiniowane za pomocą więzów (ograniczeń) w punkcie 3.3.

3.1.1 Przykład tablicy stanów lokalnych

W konflikcie między pracodawcą i pracownikami bierze udział trzech agentów $Ag = \{ag_1, ag_2, ag_3\}$ gdzie ag_1 oznacza pierwszy związek zawodowy TU1, ag_2 oznacza drugi związek zawodowy TU2 a ag_3 oznacza pracodawcę. Agenci opisują swoją sytuację w konflikcie używając następujących atrybutów $A_{ag_1} = \{a, b\}$, $A_{ag_2} = \{s, t, u\}$ i $A_{ag_3} = \{k, l, m\}$ które oznaczają odpowiednio:

a – wynagrodzenie,

b – warunki pracy,

s – płaca,

t – opieka socjalna,

u – poziom zatrudnienia,

k – zysk zakładu,

l – poziom inwestycji,

m – satysfakcja pracowników.

Dla ułatwienia założmy, że zbiór wartości jest identyczny dla wszystkich atrybutów $V_a = V_b = V_s = V_t = V_u = V_k = V_l = V_m = \{0, 1, 2\}$. Wartości z tego zbioru można interpretować jako odpowiednio niski (zerowy), średni i wysoki poziom pewnej cechy. Np. $l=2$ oznacza wysoki poziom inwestycji a $u=1$ średni poziom zatrudnienia itd. Tabela 4 przedstawia tablicę stanów lokalnych agenta ag_1 .

stany lokalne	a	b
s_1	2	2
s_2 (bieżący)	2	1
s_3	1	2
s_4	1	1
s_5	1	0

Tabela 4 Stany lokalne agenta ag_1 .

Uwaga 1 W pracy zastosowano symbol s dla oznaczenia zarówno stanu lokalnego danego agenta jak i atrybutu "płaca" agenta drugiego. Znaczenie symboli jest jednak łatwo rozróżnialne w kontekście, w którym symbol występuje.

3.1.2 Subiektywna ocena stanu lokalnego (podobieństwo stanów)

Każdy agent określa stany preferowane wśród stanów lokalnych to jest takie, które chciałby osiągnąć np. w procesie negocjacji. Podobnie stany, o których agent wie, że są niekorzystne, których zajęcie byłoby klęską agenta, znajdują się w tablicy stanów lokalnych. Agenci oceniają stany lokalne, co wyrazimy poprzez *subiektywną ocenę* stanów lokalnych. *Subiektywna ocena* wprowadza (częściowy) porządek do tablicy stanów lokalnych. Często subiektywna ocena agenta ag może być określona przez funkcję e_{ag} nazwaną *funkcją celu*, przypisującą stanom lokalnym ich ocenę subiektywną. Przykładowa *funkcja celu* używana w konflikcie między pracodawcą i pracownikami jest postaci: $e_{ag}:U_{ag} \rightarrow [0, 1]$, gdzie $[0, 1]$ oznacza przedział o końcach 0, 1 liczb rzeczywistych \mathbf{R} . Stany ocenione na 1 są najbardziej preferowane przez agenta ag , natomiast stany ocenione na 0 są przez niego nie do zaakceptowania.

Bieżący wektor informacyjny agenta ag jest informacją opisującą aktualny stan, w którym agent ag się znajduje (przy założeniu, że atrybuty A_{ag} są uporządkowane).

Na podstawie subiektywnej oceny możemy wyznaczyć również *podobieństwo* (ang. *similarity*) stanów lokalnych (liczne zastosowania podobieństwa w teorii zbiorów przybliżonych można znaleźć np. w [45], [46], [47], [29]). Dla każdego $\varepsilon > 0$, $s \in U_{ag}$ ε -sąsiedztwo stanu s definiujemy jako:

$$\tau_{ag, \varepsilon}(s) = \{s' \in U_{ag} : |e_{ag}(s) - e_{ag}(s')| \leq \varepsilon\}$$

Rodzina otoczeń $\{\tau_{ag, \varepsilon}(s)\}_{s \in U_{ag}}$ definiuje *relację tolerancji* $\tau_{ag, \varepsilon}$ w $U_{ag} \times U_{ag}$ (tzn. relację zwrotną i symetryczną) taką że: $s \tau_{ag, \varepsilon} s'$ wtedy i tylko wtedy gdy $s' \in \tau_{ag, \varepsilon}(s)$.

3.1.3 Przykład subiektywnej oceny

Stany lokalne wraz z ich subiektywną oceną dla każdego agenta biorącego udział w przykładowym konflikcie są przedstawione odpowiednio w tabelach (Tabela 5, Tabela 6, Tabela 7).

stany lokalne	a	b	e_{ag_1}
s_1	2	2	1
s_2 (bieżący)	2	1	2/3
s_3	1	2	2/3
s_4	1	1	0
s_5	1	0	0

Tabela 5 Stany lokalne agenta ag_1 i ich subiektywna ocena.

stany lokalne	s	t	u	e_{ag_2}
s_1	2	1	2	1
s_2 (bieżący)	2	2	1	1
s_3	2	1	1	2/3
s_4	1	2	1	1/3
s_5	1	1	2	1/3
s_6	1	1	1	0
s_7	2	0	1	0
s_8	0	1	2	0

Tabela 6 Stany lokalne agenta ag_2 i ich subiektywna ocena.

stany lokalne	k	l	m	e_{ag_3}
s_1	2	2	2	1
s_2	1	2	2	2/3
s_3	1	1	2	1/3
s_4	1	1	1	1/3
s_5 (bieżący)	2	0	1	0

Tabela 7 Stany lokalne agenta ag_3 z subiektywną oceną.

3.1.4 Funkcja odległości

Relacja tolerancji τ definiuje podobieństwo stanów lokalnych na podstawie ich subiektywnej oceny. Często jednak potrzebne jest wyznaczenie podobieństwa stanów na podstawie różnicy wartości atrybutów. Podobieństwo stanów z U_{ag} można zdefiniować w następujący sposób:

Zakładamy, że dla każdego atrybutu $a \in A_{ag}$ jest zdefiniowana *funkcja odległości*

$$d_a: U_{ag} \times U_{ag} \rightarrow \mathbf{R}_+$$

Na przykład $d_a(s, s') = |a(s) - a(s')|$ jeśli $V_{ag}(a) \subseteq \mathbf{R}$.

Następnie definiujemy *funkcję odległości*:

$$d: U_{ag} \times U_{ag} \rightarrow \mathbf{R}_+ \text{ jako } d(s, s') = F(d_{a_1}(s, s'), \dots, d_{a_m}(s, s')),$$

gdzie $A_{ag} = \{a_1, \dots, a_m\}$ i $F: \mathbf{R}_+^m \rightarrow \mathbf{R}_+$ jest odpowiednio dobraną funkcją np.:

$$F(r_1, \dots, r_m) = \sqrt{r_1^2 + \dots + r_m^2}.$$

Funkcja F powinna być dobrana odpowiednio do specyfiki rozważanego problemu.

Podstawą osiągnięcia konsensusu i rozwiązania konfliktu jest gotowość agentów do zmiany zajmowanego stanu i być może rezygnacji z części posiadanych zasobów. Gotowość ta jest podstawą do zdefiniowania *bliskiego sąsiedztwa* danego stanu – zakładamy, że agent jest gotowy zaakceptować stany z *bliskiego sąsiedztwa*. *Bliskość* stanów definiujemy poprzez funkcję odległości w następujący sposób.

Dwa stany s i s' są sobie ϵ_{ag} -bliskie wtedy i tylko wtedy, gdy $d(s, s') < \epsilon_{ag}$, gdzie ϵ_{ag} jest danym progiem *bliskości* dla ag . Konsekwentnie *bliskie sąsiedztwo* stanu s w stopniu ϵ_{ag} definiujemy jako $\{s': d(s, s') < \epsilon_{ag}\}$.

3.1.4.1 Przykład bliskiego sąsiedztwa

Niech $\epsilon_{ag_3} = 2/3$. Bliskie sąsiedztwo stanu lokalnego s_2 w stopniu $2/3$ jest zaprezentowane w tabelicy – Tabela 8. Dla przykładu obliczmy odległość stanów s_2 i $s_{2,1}$ przyjmując funkcję $F(v_1, v_2, v_3) = 1/3(v_1 + v_2 + v_3)$.

$$d(s_2, s_{2,1}) = F(d_k(s_2, s_{2,1}), d_l(s_2, s_{2,1}), d_m(s_2, s_{2,1})) = 1/3(1+0+0) = 1/3$$

stany lokalne	k	l	m
s_2	1	2	2
$s_{2,1}$	2	2	2
$s_{2,2}$	1	1	2
$s_{2,3}$	1	2	1

Tabela 8 Bliskie sąsiedztwo stanu s_2 w stopniu określonym przez parametr $2/3$.

3.1.5 Lokalne zbiory celów

Funkcja celu wprowadza częściowy porządek w zbiorze stanów lokalnych, tak że można znaleźć maksymalny element (elementy) (z najwyższą oceną) i minimalny element

(elementy). Maksymalne elementy można interpretować jako cele danego agenta, to znaczy, są to takie stany do uzyskania, których agent dąży np. w procesie negocjacji.

Zbiór celów T_{ag} agenta ag definiujemy zatem jako:

$$T_{ag} = \{s \in U_{ag} : e_{ag}(s) > \mu_{ag}\},$$

gdzie μ_{ag} jest *poziomem akceptacji*, wybranym subiektywnie przez agenta ag (ocena, które stany są preferowane jest subiektywna dla każdego agenta).

3.1.6 Przykład zbiorów celu, znajdowanie opisów zbiorów z pomocą reguł decyzyjnych

Niech dla rozważanej sytuacji konfliktowej, minimalny poziom akceptacji dla wszystkich agentów będzie taki sam i wynosi $1/3$. Zbiory celów agentów będą więc następujące (zob. Przykład 3.1.3): $T_{ag_1} = \{s_1, s_2, s_3\}$, $T_{ag_2} = \{s_1, s_2, s_3\}$, $T_{ag_3} = \{s_1, s_2\}$.

Zbiory celów mogą być również przedstawione w postaci formuł Boolowskich. Zmiennymi w tych formułach są pary (*atrybut, wartość*). W celu znalezienia formuły opisującej zbiór celów dla danego agenta zmieniamy jego tablicę stanów lokalnych w tablicę decyzyjną w ten sposób, że atrybut decyzyjny ma wartość 1 dla stanów należących do zbioru celów i 0 dla pozostałych stanów. Następnie, dla tak określonej tablicy decyzyjnej znajdujemy reguły decyzyjne [29], np. minimalne ze względu na liczbę atrybutów występujących po lewej stronie reguły. Suma logiczna (dyzjunkcja) lewych stron reguł o decyzji 1 jest poszukiwaną formułą opisującą zbiór celów agenta. Dla przykładu znajdziemy formułę opisującą zbiór celów dla agenta ag_1 .

stany lokalne	a	b	decyzja d
s_1	2	2	1
s_2	2	1	1
s_3	1	2	1
s_4	1	1	0
s_5	1	0	0

Tabela 9 Tablica decyzyjna stanów lokalnych agenta ag_1 .

Tablicę decyzyjną (Tabela 9) przekształcamy w *tablicę rozróżnialności* stanów ze zbioru celów s_1, s_2, s_3 z pozostałymi stanami (Tabela 10).

stany lokalne	s_1	s_2	s_3
s_4	a b	a	b
s_5	a b	a b	b

Tabela 10 Tablica rozróżnialności stanów lokalnych agenta ag_1 .

Z tablicy rozróżnialności obliczamy *reduktę* [29] – odpowiadają one implikantom pierwszym formuły rozróżnialności:

$$f_{A_{ag_1}}(a, b) = (a \vee b) \wedge a \wedge b \wedge (a \vee b) \wedge (a \vee b) \wedge b \equiv a \wedge b$$

Aby otrzymać reguły o minimalnej liczbie atrybutów po lewej stronie należy odróżnić każdy stan z osobna (rozbić tablicę rozróżnialności na trzy tablice). Pytamy się wtedy o minimalną formułę opisującą zadaną decyzję. Mamy wtedy następujące formuły rozróżnialności:

$$f^{s^1}_{A_{ag_1}}(a, b) = (a \vee b) \wedge (a \vee b) \equiv a \vee b$$

$$f^{s^2}_{A_{ag_1}}(a, b) = a \wedge (a \vee b) \equiv a$$

$$f^{s^3}_{A_{ag_1}}(a, b) = b \wedge b \equiv b$$

Co daje następujący opis klasy decyzyjnej dla decyzji $d=1$:

$$a_2 \vee b_2 \rightarrow d_1$$

Aby jednoznacznie sklasyfikować nowy (nieznany dotąd) obiekt musimy znaleźć dopełnienie wyznaczonej klasy decyzyjnej, czyli opis klasy decyzyjnej decyzji $d=0$. Sposób postępowania jest identyczny jak dla decyzji $d=1$. Ostatecznie otrzymujemy:

$$b_0 \vee (a_1 \wedge b_1) \rightarrow d_0$$

Uwaga 2 Można zauważyć, że np. obiekt opisany przez $a_2 \wedge b_0$ nie może być jednoznacznie sklasyfikowany do którejś z klas wyznaczonych przez atrybut decyzyjny, ponieważ zgodnie z przedstawionym opisem zostanie on rozpoznany jako obiekt zarówno z klasy 0 jak i z klasy 1. Aby temu zapobiec trzeba wprowadzić mechanizm rozstrzygania konfliktów między regułami głosującymi na różne decyzje przy rozpoznawaniu tego samego obiektu [63].

Formuła opisująca zbiór celów dla agenta ag_1 jest więc następująca $t_{ag_1} = a_2 \vee b_2$.

W podobny sposób znajdujemy formuły opisujące zbiory celów dla pozostałych agentów.

$$t_{ag_2} = (s_2 \wedge t_1) \vee (s_2 \wedge t_2) \vee (s_2 \wedge u_2)$$

$$t_{ag_3} = l_2 \vee (k_2 \wedge m_2)$$

Uwaga 3 Przedstawiony powyżej sposób wyznaczania opisu danego zbioru jest oparty na teorii zbiorów przybliżonych i stosowany w wielu pracach [45], [46]. Wyznaczona formuła opisuje również obiekty (patrz rozdział 12.4) nie występujące w tabelach. Ma to szczególne znaczenie przy opisie sytuacji konfliktowych, gdzie mamy do czynienia z zazwyczaj niepełną i niepewną informacją. Na przykład w bardziej złożonej sytuacji konfliktowej nierealnym byłoby założenie, że agent jest w stanie wyspecyfikować wszystkie możliwe stany lokalne przez ich wyliczenie.

Uwaga 4 W dalszej części pracy znak koniunkcji i otaczające jego argumenty nawiasy będą pomijane (zgodnie z zasadą, że operator koniunkcji wiąże silniej od operatora dyzjunkcji). Dlatego wyrażenie $\alpha \wedge \beta \vee \gamma \wedge \delta$ będzie rozumiane jako $(\alpha \wedge \beta) \vee (\gamma \wedge \delta)$.

Uwaga 5 Choć zastosowano te same oznaczenie, w zależności od kontekstu można odróżnić $a=v$, które oznaczają $\{ag \in U_{ag}: a(ag)=v\}$ od zmiennej boolowskiej $(a=v)^*$ odpowiadającej $a=v$ (i też oznaczanej przez $a=v$ lub a_v). W całym dokumencie ograniczono się do wartościowań val spełniających następujące warunki:

$$\begin{cases} val(a_v \wedge a_{v'}) = 0 & \text{dla } v \neq v' \\ val(\bigvee_{v \in V_a} a_v) = 1 \end{cases}$$

Formuła boolowska f jest sprzeczna (fałszywa) ($f \equiv 0$) wtedy i tylko wtedy, gdy dla każdego val spełniającego powyższe warunki $val(f)=0$. Odpowiednio formuła boolowska f jest niesprzeczna wtedy i tylko wtedy, gdy istnieje takie val spełniające powyższe warunki, że $val(f)=1$.

3.2 Sytuacje

Rozważmy zbiór Ag składający się z n agentów ag_1, \dots, ag_n . Sytuacją dla Ag nazywamy każdy element iloczynu kartezjańskiego:

$$S_{Ag} = \prod_{i=1}^n INF^*(ag_i)$$

gdzie $INF^*(ag_i)$ jest zbiorem wszystkich możliwych informacji agenta ag_i , i jest zdefiniowany jako:

$$INF^*(ag) = \{f : A_{ag} \rightarrow \bigcup_{a \in A_{ag}} V_a(ag) : f(a) \in V_a(ag) \text{ dla } a \in A_{ag}\}$$

Sytuacja S_{A_g} odpowiadająca globalnemu stanowi $S=(s_1, \dots, s_n) \in U_{ag_1} \times \dots \times U_{ag_n}$ jest zdefiniowana przez $(Inf_{A_{ag_1}}(s_1), \dots, Inf_{A_{ag_n}}(s_n))$.

Przykład sytuacji w rozważanym konflikcie prezentuje Tabela 11.

Sytuacja	a	b	s	t	u	k	l	m
S_b	2	1	2	2	1	2	0	1

Tabela 11 Przykładowa sytuacja.

3.3 Ograniczenia (więzy)

Ograniczenia wprowadzają zależności między stanami lokalnymi agentów. Bez tych zależności każdy agent mógłby zająć dowolny stan (np. stan ze swego zbioru celów). Agenci byłiby od siebie niezależni i nie byłoby konfliktu. Ograniczenia wynikają ze skończoności zasobów w danej sytuacji – różnego typu zasoby mogą być brane pod uwagę np. woda na Wzgórzach Golan [43], międzynarodowa pozycja, wszystko co jest ważne dla agentów. Ograniczenia wyrażają, które stany lokalne mogą ze sobą współistnieć, formalnie zaś definiują podzbiór zbioru sytuacji. Ograniczenia zawężają zbiór wszystkich możliwych sytuacji do *sytuacji osiągalnych* spełniających ograniczenia.

W dalszej części pracy będziemy rozważać tylko osiągalne sytuacje nazywając je krótko sytuacjami.

3.3.1 Przykład ograniczeń

Następujące zależności zawężają zbiór sytuacji i są ograniczeniami w naszym konflikcie pomiędzy pracodawcą i pracownikami. Nazwy atrybutów oznaczają zmienne i przyjmują wartości ze zbioru wartości odpowiadającego atrybutu. Stałe w zależnościach zostały dobrane eksperymentalnie aby wyrazić związki między różnymi atrybutami.

1. $a > 0$ (poziom nakładów na pracowników musi być co najmniej średni)
2. $u > 0$ (poziom zatrudnienia musi być co najmniej średni)

3. $l + m \geq u$ (poziom zatrudnienie zależy od poziomu inwestycji i satysfakcji pracowników)
4. $2 + a \geq s + t$ (na wynagrodzenie składają się płaca i warunki socjalne)
5. $2 + m = a + b$ (satysfakcja pracowników zależy od wynagrodzenia i warunków pracy)
6. $3 * k \geq a + l + s + t$ (podział zysku – najbardziej podstawowy przypadek – dochód zakładu jest dzielony na wszystkie wydatki)

Tak jak wspomniano, ograniczenia zawężają zbiór sytuacji do sytuacji, w których stany lokalne agentów mogą ze sobą współistnieć. Na przykład, sytuacja $a=2, b=2, s=2, t=2, u=2, k=2, l=2, m=2$ nie jest osiągalna ze względu na ograniczenie 6.

Powyższe ograniczenia mogą być opisane za pomocą odpowiadających formuł boolowskich odpowiednio $f_{\varphi_1}, f_{\varphi_2}, f_{\varphi_3}, f_{\varphi_4}, f_{\varphi_5}, f_{\varphi_6}$. Koniunkcja $f_{\varphi} = f_{\varphi_1} \wedge f_{\varphi_2} \wedge f_{\varphi_3} \wedge f_{\varphi_4} \wedge f_{\varphi_5} \wedge f_{\varphi_6}$ definiuje, w zbiorze wszystkich sytuacji, podzbiór wszystkich osiągalnych sytuacji w naszym przykładowym konflikcie spełniających f_{φ} . Dokładniej, sytuacja osiągalna musi być niesprzeczna przynajmniej z jednym z implikantów pierwszych formuły f_{φ} . Dla przykładu znajdziemy formułę f_{φ_1} . Ponieważ zbiorem wartości atrybutu a jest $\{0, 1, 2\}$ to formuła ta jest następującej postaci: $f_{\varphi_1} = a_1 \vee a_2$.

Uwaga 6 Zadanie wyznaczenia rozwiązań z zależności jest zadaniem z dziedziny "programowania matematycznego" (ang. constraints programming CSP) [50], [6]. Polega ono ogólnie na wyznaczeniu wszystkich rozwiązań spełniających podane warunki (ograniczenia). Autor przygotował do tego celu specjalny komputerowy moduł programowy.

Do dalszych obliczeń wygodniej będzie nie redukować formuły f_{φ} (patrz punkt 8.2.2) do normalnej postaci dyzjunkcyjnej (w postaci sumy implikantów pierwszych).

3.4 Ocena sytuacji

Podobnie jak stan lokalny, sytuacja globalna podlega ocenie. Ocena ta będzie stanowiła odniesienie przy szukaniu konsensusu (patrz rozdział 5.1). Może ona odzwierciedlać preferencje agentów (subiektywną ocenę stanów) i wtedy konsensus będzie polegał na znalezieniu sytuacji najbardziej preferowanej przez maksymalną liczbę agentów (rozdział 5.3). Odpowiada to sytuacji, gdy agenci dążą do zajęcia jak najlepszego stanu nie dbając o *dobro wspólne*. Jak wykazuje jednak doświadczenie negocjatorów stabilny konsensus można osiągnąć tylko wtedy gdy agenci biorą pod uwagę dobro wspólne [34]. Innym więc

sposobem oceny sytuacji jest osąd eksperta, arbitra. Przykładem takiego eksperta w międzynarodowych konfliktach zbrojnych może być Organizacja Narodów Zjednoczonych.

3.4.1 Obiektywna ocena sytuacji

Niech będzie dana funkcja $q: S_{Ag} \rightarrow [0, 1]$, nazwana *funkcją jakości*, która każdej sytuacji osiągalnej przypisuje jej ocenę.

Zbiór sytuacji, których ocena jest nie mniejsza od zadanego dopuszczalnego poziomu t będziemy nazywali *zbiorem sytuacji t-dobrych* i będziemy oznaczali $Score_{Ag}(t)$. Dokładniej:

$$Score_{Ag}(t) = \{S \in S_{Ag} : q(S) \geq t\}$$

3.4.1.1 Przykład sytuacji dobrych

Tabela 12 prezentuje sytuacje ocenione przez eksperta dla konfliktu między pracodawcą a pracownikami. Niech wymagany poziom jakości (decyzja 1) będzie $t=2/3$.

Sytuacje	<i>a</i>	<i>b</i>	<i>s</i>	<i>t</i>	<i>u</i>	<i>k</i>	<i>l</i>	<i>m</i>	$q(S)$	decyzja
S_1	1	2	1	2	1	2	2	2	1	1
S_2	1	2	2	2	2	2	2	1	1	1
S_3	1	2	1	1	1	2	2	1	1	1
S_4	1	1	2	1	1	1	1	1	2/3	1
S_5	1	2	1	2	1	1	2	2	2/3	1
S_6	1	2	1	1	2	2	2	1	2/3	1
S_7	1	2	1	2	1	1	2	2	1/3	0
S_8	1	2	1	2	1	2	1	2	1/3	0
S_9	1	2	1	2	1	2	2	1	1/3	0
S_{10}	1	2	2	2	2	2	2	0	1/3	0
S_{11}	1	2	2	2	2	1	2	1	0	0
S_{12}	1	2	2	2	2	2	1	1	1/3	0
S_{13}	1	2	1	1	1	2	2	0	1/3	0
S_{14}	1	2	1	1	1	2	1	1	0	0
S_{15}	1	2	1	1	1	0	2	1	0	0
S_{16}	1	1	2	1	1	0	1	1	0	0
S_{17}	1	1	2	1	1	1	0	1	0	0
S_{18}	1	1	2	1	1	1	1	0	0	0
S_{19} (bieżąca)	1	2	2	2	1	2	0	1	0	0
S_{20}	1	2	1	2	1	0	2	2	0	0
S_{21}	1	2	1	2	1	1	0	2	0	0
S_{22}	1	2	1	2	1	1	2	1	0	0

Sytuacje	<i>a</i>	<i>b</i>	<i>s</i>	<i>t</i>	<i>u</i>	<i>k</i>	<i>l</i>	<i>m</i>	<i>q(S)</i>	decyzja
S_{23}	1	2	1	1	2	2	1	1	1/3	0
S_{24}	1	2	1	1	2	2	2	0	0	0
S_{25}	1	2	1	1	2	0	2	1	0	0
S_{26}	2	2	1	2	1	2	2	2	1/3	0
S_{27}	2	1	2	2	2	2	2	1	1/3	0
S_{28}	2	0	1	1	1	2	2	1	0	0

Tabela 12 Obiektywna ocena sytuacji (tablica decyzyjna).

Znajdujemy opis w postaci formuły dla zbioru sytuacji dobrych. Sposób postępowania jest identyczny jak przy poszukiwaniu opisu zbiorów celów (Przykład 3.1.6).

$$a_1k_2l_2m_2 \vee a_1s_2k_2l_2m_1 \vee b_2s_2k_2l_2m_1 \vee a_1u_2k_2l_2m_1 \vee b_2u_2k_2l_2m_1 \vee a_1t_1k_2l_2m_1 \vee b_2t_1k_2l_2m_1 \vee k_1l_1m_1 \vee k_1l_2m_2 \vee t_1u_2k_2l_2m_1 \vee s_1u_2k_2l_2m_1 \leftrightarrow d_1$$

Uwaga 7 Oczywiście ekspert nie jest w stanie ocenić wszystkich sytuacji osiągalnych w danym konflikcie. Podkreślmy więc, że zastosowana metoda pozwala dzięki wnioskowaniu indukcyjnemu aproksymować ocenę eksperta na wszystkie sytuacje z decyzją 1 (szczegółowy sposobu wyznaczania minimalnych reguł decyzyjnych są przedstawione w rozdziale 12.4).

Uwaga 8 Zaproponowane w pracy algorytmy poszukiwania konsensusu należy traktować jako wspomaganie decyzji. Należy więc poddać weryfikacji eksperta propozycje sytuacji obiektywnie niekonfliktowych (rozdział 5.1). Jeżeli otrzymana nowa sytuacja (propozycja konsensusu) zostaje odrzucona przez eksperta warto ją dodać do tablicy decyzyjnej i powtórnie wyznaczyć opis zbioru sytuacji dobrych. Możliwość uczenia się na podstawie przykładów jest ważną cechą zastosowanej metody [21].

3.4.2 Preferencje agentów

Innym sposobem oceny sytuacji jest przeniesienie ocen stanów lokalnych (preferencji lokalnych) na poziom sytuacji. Ocena ta wyraża preferencje agentów (dlatego jest nazwana *globalną oceną preferencji*) i nie uwzględnia dobra wspólnego. Dokładniej *globalną funkcję preferencji* sytuacji osiągalnej S odpowiadającej globalnemu stanowi $\bar{s} = (s_1, \dots, s_n)$ definiujemy w następujący sposób:

$$p(\bar{s}) = F(e_{ag_1}(s_1), e_{ag_2}(s_2), \dots, e_{ag_n}(s_n))$$

gdzie $\bar{s} = (s_1, s_2, \dots, s_n)$ i F jest odpowiednio dobraną funkcją na przykład $F(r_1, \dots, r_m) = \sum_{i=1}^m r_i$

[28]. Jeżeli znane jest znaczenie poszczególnych agentów (porównaj punkt 6.5.1) to warto je

uwzględnić w funkcji preferencji np. poprzez wprowadzenie wag tzn.: $F(r_1, \dots, r_m) = \sum_{i=1}^m w_i r_i$,

gdzie w_i jest znaczeniem (waga) i -tego agenta.

Odpowiednio zbiór wszystkich preferowanych sytuacji zdefiniujemy przez:

$$Spref_{Ag}(h) = \{S: p(S) \geq h\}, \text{ gdzie } h \text{ jest wybranym poziomem preferencji.}$$

Uwaga 9 Zaproponowana i używana w dalszej części pracy funkcja F jest bardzo prosta. Zakłada, że agenci używają tej samej skali do wyrażenia lokalnych preferencji. Można jednak zastosować inną funkcję odpowiednią dla danej sytuacji, w szczególności nieliniową *globalną ocenę preferencji* można aproksymować np. z pomocą reguł decyzyjnych. Uwzględnienie nieliniowości przy analizie problemów występujących w sytuacjach konfliktowych sprawia bardzo duże trudności jeżeli analizę tę chciałoby się prowadzić metodami analitycznymi [28]. Proponowane podejście pozwala operować zbiorem reguł opisujących zbiory sytuacji preferowanych zarówno w przypadku liniowym jak i nieliniowym.

3.5 System z ograniczeniami

System wieloagentowy, w którym są zdefiniowane stany lokalne dla każdego agenta i sytuacje globalne spełniające ograniczenia będziemy nazywać *systemem z ograniczeniami*, i będziemy go oznaczać M_{Ag} .

KONFLIKTY

W poniższym rozdziale zdefiniujemy pojęcie konfliktu w systemie z ograniczeniami. W takim systemie konflikt może wystąpić zarówno na poziomie lokalnym jak i globalnym. Jak zobaczymy konflikt lokalny jest związany z niską samooceną bieżącego stanu agenta. Konflikt globalny wskazuje na zaistnienie sytuacji niekorzystnej z punktu widzenia dobra wspólnego lub w związku z tym, że dana sytuacja nie jest preferowana przez większość agentów.

4.1 Konflikt lokalny

Agent ag w stanie s jest w *konflikcie lokalnym* w stopniu co najmniej ε wtedy i tylko wtedy gdy s nie należy do ε -sąsiedztwa s' , dla żadnego s' należącego do zbioru *celów* agenta ag . Można to wyrazić również stwierdzeniem, że stan s nie należy do ε -otoczenia zbioru celów T_{ag} tzn.

$$s \notin \bigcup_{s' \in T_{ag}} \tau_{ag, \varepsilon}(s') \text{ gdzie } \tau_{ag, \varepsilon}(s') = \{s'' : s'' \tau_{ag, \varepsilon} s'\}.$$

Konflikt lokalny wynika więc z niskiej subiektywnej oceny bieżącego stanu lokalnego danego agenta. *Natężenie* konfliktu lokalnego agenta ag w stanie $s \in U_{ag}$ możemy wyrazić za pomocą następującego współczynnika:

$$Cl_{ag}^{\varepsilon}(s) = \begin{cases} f_{ag}(s) - \varepsilon, & \text{dla } f_{ag}(s) > \varepsilon \\ 0, & \text{w przeciwnym razie} \end{cases}$$

gdzie ε jest parametrem konfliktu lokalnego wyrażającym, że dopuszczalny stopień odchylenia oceny stanu aktualnego od oceny pewnego stanu ze zbioru stanów docelowych T_{ag} nie może być większy niż ε . Funkcja $f_{ag}(s)$, jest funkcją odległości stanu s od zbioru celów

agenta ag tzn. $f_{ag}(s) = \min\{e_{ag}(s) - e_{ag}(s') : s' \in T_{ag}\}$, gdzie $e_{ag}(s)$ jest subiektywną oceną agenta ag stanu s .

Przykład 4.1

Sprawdźmy czy istnieje konflikt lokalny w omawianej sytuacji konfliktowej zakładając, że $\varepsilon=0$ tzn. stan bieżący musi należeć do zbioru celów. Ponieważ $Cl_{ag_1}^\varepsilon(s_2) = 0$ – brak konfliktu lokalnego, stan bieżący agenta ag_1 należy do zbioru celów. Podobnie $Cl_{ag_2}^\varepsilon(s_2) = 0$. Natomiast $Cl_{ag_3}^\varepsilon(s_5) = |0 - 2/3| - 0 = 2/3$ agent ag_3 jest w konflikcie lokalnym o natężeniu $2/3$.

4.2 Subiektywny konflikt globalny

Sytuacja S jest *subiektywnie konfliktowa dla zbioru agentów Ag w stopniu t* wtedy i tylko wtedy, gdy przy zadanym poziomie preferencji t , sytuacja S nie należy do zbioru sytuacji preferowanych $S_{pref_{Ag}}(t)$. Gdy sytuacja bieżąca jest subiektywnie konfliktowa dla Ag to agenci należący do zbioru Ag są w *subiektywnym globalnym konflikcie*.

Różnica zadanego poziomu preferencji t i wartości *funkcji preferencji* $p(S)$ może służyć jako *współczynnik natężenia subiektywnego konfliktu globalnego*:

$$Cp_{Ag}^t(S) = \begin{cases} t - p(S) & \text{dla } t > p(S) \\ 0, & \text{w przeciwnym razie} \end{cases}$$

Subiektywny konflikt globalny wynika z niezaspokojonych preferencji agentów.

Przykład 4.2

Wartość funkcji preferencji p w bieżącej sytuacji S_{19} (Tabela 12) wynosi $p(S_{19}) = 2/3 + 2/3 + 0 = 4/3$. Niech wymagany poziom preferencji $t = 2$. Agenci są więc w subiektywnym konflikcie globalnym o natężeniu $Cp_{Ag}^t(S_{19}) = 2/3$.

4.3 Obiektywny konflikt globalny

Sytuacja S jest *obiektywnie konfliktowa* dla Ag wtedy i tylko wtedy gdy przy zadanym poziomie jakości t , sytuacja S nie należy do zbioru *sytuacji dobrych* $Score_{Ag}(t)$. Gdy sytuacja bieżąca jest obiektywnie konfliktowa dla Ag to agenci ze zbioru Ag są w *obiektywnym globalnym konflikcie*.

Natężenie obiektywnego konfliktu globalnego można wyrazić z pomocą współczynnika $Cg_{Ag}^t(S)$:

$$Cg_{Ag}^t(S) = \begin{cases} t - q(S) & \text{dla } t > q(S) \\ 0 & \text{w przeciwnym razie} \end{cases}$$

gdzie $q(S)$ jest funkcją jakości a t zadanym poziomem jakości sytuacji.

Przykład 4.3

Niech zadany poziom jakości wynosi $t=2/3$ dla omawianego konfliktu. W sytuacji bieżącej S_{19} istnieje obiektywny konflikt globalny pomiędzy agentami ag_1 , ag_2 , ag_3 o natężeniu $Cg_{Ag}^t(S_{19}) = 2/3 - 0 = 2/3$.

4.4 Różnorodność konfliktów

Powyżej zostało zdefiniowane pojęcie konfliktu w modelu sytuacji konfliktowych. Jest ono podstawą do poszukiwania konsensusu – rozwiązania konfliktu. Jak można zauważyć samo pojęcie konfliktu nie jest jednoznaczne. Konflikt może występować na płaszczyźnie lokalnej i globalnej. W zależności też od zastosowanych kryteriów może być subiektywny lub obiektywny. (Oczywiście możliwe jest również zastosowanie innych kryteriów.) Idealnym rozwiązaniem konfliktu jest znalezienie takich sytuacji, w których nie występuje żaden z wyżej wymienionych konfliktów – w której agenci zajmują stany preferowane i sytuacja jest obiektywnie dobra. W następnym rozdziale wskażemy sposoby poszukiwania rozwiązania dla różnych typów konfliktów.

ANALIZA PROBLEMU KONSENSUSU

Przedstawiony powyżej model daje przede wszystkim możliwość zrozumienia i analizy różnego rodzaju konfliktów. Jest podstawą rozwiązywania konfliktów, szukania konsensusu [34]. Podobnie jak w rzeczywistych sytuacjach konfliktowych konsensus można znaleźć na różnych poziomach i pod pewnymi warunkami. W tym rozdziale przeanalizujemy szczegółowo problem konsensusu, pokażemy też algorytmy poszukiwania sytuacji niekonfliktowych. W algorytmach zastosowano metody znane z teorii zbiorów przybliżonych i wnioskowanie boolowskie.

5.1 Konsensus na poziomie globalnym (obiektywna ocena)

Problem konsensusu na poziomie globalnym może być zdefiniowany w następujący sposób.

DANE WEJŚCIOWE

System z ograniczeniami M_{Ag} zdefiniowany w rozdziale 3.

t – akceptowany poziom jakości sytuacji globalnych.

WYNIK

Zbiór Sc wszystkich sytuacji niekonfliktowych tj. takich, że $Cg_{Ag}^t(S') = 0$ dla $S' \in Sc$.

ALGORYTM

Należy przeanalizować wszystkie sytuacje osiągalne i znaleźć te, na których wartość funkcji jakości jest nie mniejsza od zadanego poziomu t .

Znalezienie rozwiązania polega na znalezieniu najpierw formuły opisującej zbiór sytuacji dobrych $Score_{Ag}(t)$ dla poziomu jakości t – formuła f_C . Następnie formułę tę należy zweryfikować (przez użycie koniunkcji) z formułą opisującą ograniczenia f_φ z uwagi na to, że

formuła f_C może zawierać sytuacje nieosiągalne. Problem możemy więc zakodować za pomocą formuły:

$$f_N = f_C \wedge f_\varphi$$

Implikanty pierwsze tej formuły opisują wszystkie obiektywnie niekonfliktowe sytuacje osiągalne w tym sensie, że sytuacja odpowiadająca dowolnemu implikantowi pierwszemu tej formuły jest sytuacją obiektywnie niekonfliktową.

Znalezienie konsensusu na poziomie globalnym nie daje pełnego rozwiązania z uwagi na to, że nie uwzględnione są preferencje agentów. Proponowane sytuacje powinny być zaakceptowane przez większość agentów. W następnym punkcie będziemy poszukiwali takich rozwiązań, które likwidują także konflikty lokalne.

Przykład 5.1

Formuła f_C opisuje sytuacje dobre na podstawie oceny eksperta. Sposób wyznaczenia tej formuły dla poziomu $t=1/3$ został opisany w rozdziale 3.4.1.1. Natomiast sposób przekształcenia ograniczeń do postaci formuły przedstawiono w 3.3.1. Dla rozważanego przykładu konflikt ma 21 rozwiązań (które opisuje formuła f_N) – sytuacje niekonfliktowe (w ocenie obiektywnej). Zostały one obliczone za pomocą programu przygotowanego przez autora.

$$\begin{aligned} f_N = & a_1 b_0 s_2 t_1 u_1 k_2 l_2 m_1 \vee a_1 b_0 s_2 t_1 u_2 k_2 l_2 m_1 \vee a_1 b_0 s_1 t_2 u_2 k_2 l_2 m_1 \vee a_1 b_1 s_2 t_1 u_1 k_2 l_2 m_1 \vee a_1 b_1 s_2 t_1 u_2 k_2 l_2 m_1 \vee \\ & a_1 b_1 s_1 t_2 u_2 k_2 l_2 m_1 \vee a_1 b_2 s_2 t_1 u_1 k_2 l_2 m_1 \vee a_1 b_2 s_2 t_1 u_2 k_2 l_2 m_1 \vee a_1 b_2 s_1 t_2 u_2 k_2 l_2 m_1 \vee a_1 b_0 s_2 t_1 u_1 k_2 l_2 m_2 \vee \\ & a_1 b_0 s_2 t_1 u_2 k_2 l_2 m_2 \vee a_1 b_0 s_1 t_2 u_1 k_2 l_2 m_2 \vee a_1 b_0 s_1 t_2 u_2 k_2 l_2 m_2 \vee a_1 b_1 s_2 t_1 u_1 k_2 l_2 m_2 \vee a_1 b_1 s_2 t_1 u_2 k_2 l_2 m_2 \vee \\ & a_1 b_1 s_1 t_2 u_1 k_2 l_2 m_2 \vee a_1 b_1 s_1 t_2 u_2 k_2 l_2 m_2 \vee a_1 b_2 s_2 t_1 u_1 k_2 l_2 m_2 \vee a_1 b_2 s_2 t_1 u_2 k_2 l_2 m_2 \vee a_1 b_2 s_1 t_2 u_1 k_2 l_2 m_2 \vee \\ & a_1 b_2 s_1 t_2 u_2 k_2 l_2 m_2 \end{aligned}$$

Uwaga 10 Przygotowany przez autora program generuje wszystkie implikanty pierwsze formuł boolowskich. Danymi wejściowymi programu są tablice decyzyjne i formuły wygenerowane na podstawie ograniczeń. Program znajduje koniunkcję formuł i wyprowadza implikanty pierwsze. Szczegóły algorytmów redukcji formuł boolowskich opisano w rozdziale 8.2.

5.2 Konsensus na poziomie globalnym i lokalnym

Zdefiniujemy teraz problem konsensusu uwzględniając preferencje lokalne agentów.

DANE WEJŚCIOWE

System z ograniczeniami M_{Ag} zdefiniowany w rozdziale 3.

t – akceptowany poziom jakości sytuacji globalnych.

WYNIK

Zbiór wszystkich takich sytuacji niekonfliktowych, których obiektywna ocena jest co najmniej t i w których nie występuje konflikt lokalny dla żadnego z agentów. Problem polega na szukaniu konsensusu lepszego niż poprzednio – każda skonstruowana nowa sytuacja, zaproponowana jako rozwiązanie konfliktu musi dodatkowo spełniać preferencje agentów.

ALGORYTM

Algorytm polega na weryfikacji sytuacji dobrych (zbiór $Score_{Ag}(t)$) względem preferencji agentów (zbiory celów) i oczywiście ograniczeń. Problem opisuje następująca formuła:

$$f = \bigwedge_{ag \in Ag} t_{ag} \wedge f_C \wedge f_\varphi$$

gdzie formuła t_{ag} opisuje zbiór celów agenta ag , natomiast koniunkcję formuł $f_C \wedge f_\varphi$ rozważaliśmy w rozdziale 5.1 – opisuje ona problem konsensusu na poziomie globalnym z parametrem t .

Przykład 5.2

Tworzenie formuł f_C i f_φ zostało opisane odpowiednio w punktach 3.4.1.1 i 3.3.1. Pokazaliśmy też jak otrzymać formuły opisujące zbiory celów agentów (punkt 3.1.6). Są one następujące:

$$t_{ag_1} = a_2 \vee b_2$$

$$t_{ag_2} = s_2 t_1 \vee s_2 t_2 \vee s_2 u_2$$

$$t_{ag_3} = l_2 \vee k_2 m_2$$

Formuła f dla naszego przykładowego konfliktu będzie miała postać (ze względu na wielkość nie rozwinęliśmy tutaj formuł f_C i f_ϕ):

$$f = (a_2 \vee b_2) \wedge (s_2 t_1 \vee s_2 t_2 \vee s_2 u_2) \wedge (l_2 \vee k_2 m_2) \wedge f_C \wedge f_\phi$$

Po redukcji formuła f ma następującą postać (wykorzystano komputerowy program autora):

$$f = a_1 b_2 s_2 t_1 u_1 k_2 l_2 m_1 \vee a_1 b_2 s_2 t_1 u_2 k_2 l_2 m_1 \vee a_1 b_2 s_2 t_1 u_1 k_2 l_2 m_2 \vee a_1 b_2 s_2 t_1 u_2 k_2 l_2 m_2$$

Wynik ten oznacza, że następujące cztery sytuacje spełniają zadane warunki konsensusu (Tabela 13).

sytuacje	a	b	s	t	u	k	l	m
S_1	1	2	2	1	1	2	2	1
S_2	1	2	2	1	2	2	2	1
S_3	1	2	2	1	1	2	2	2
S_4	1	2	2	1	2	2	2	2

Tabela 13 Tablica sytuacji niekonfliktowych.

Zaproponowane sytuacje można interpretować jako konieczność rezygnacji związków zawodowych z pewnych już uzyskanych warunków płacowych ($a=2, t=2$) na korzyść lepszej kondycji zakładu – większych nakładów inwestycyjnych ($l=2$).

Uwaga 11 Warto zauważyć, że nie wszystkie stany lokalne będące rozwiązaniem konfliktu zostały uwzględnione w tablicach lokalnych agentów. Przedstawiane tu podejście umożliwia poszukiwanie konsensusu niesprzecznego z opisem zbiorów celów uzyskanym z tablic stanów lokalnych w postaci reguł decyzyjnych dla każdego agenta.

5.2.1 Konsensus w bliskim sąsiedztwie celów agentów

W tym przypadku weźmiemy dodatkowo pod uwagę bliskie sąsiedztwo celów agentów. Często zdarza się, że cele agentów są zbyt "wygórowane" i nie można znaleźć żadnego rozwiązania. Poszukując konsensusu w bliskim sąsiedztwie celów agentów mamy szansę znaleźć inne rozwiązania możliwe do zaakceptowania przez agentów. Problem konsensusu w bliskim sąsiedztwie celów agentów można zdefiniować w następujący sposób.

DANE WEJŚCIOWE

System z ograniczeniami M_{Ag} zdefiniowany w rozdziale 3.

t – akceptowany poziom jakości sytuacji globalnych.

Parametr bliskości ε .

WYNIK

Zbiór wszystkich sytuacji bez konfliktu lokalnego, dla których obiektywna ocena jest co najmniej t . Nowe sytuacje mogą spełniać preferencje agentów rozszerzone o stany z bliskiego sąsiedztwa zbiorów celów agentów.

ALGORYTM

Algorytm jest podobny do poprzedniego (punkt 5.2), przy czym dla każdego agenta konstruowany jest nowy zbiór celów na podstawie bliskiego sąsiedztwa stanów ze zbiorów celów. Dokładnie implikanty pierwsze formuły f' dają wszystkie rozwiązania naszego problemu.

$$f' = \bigwedge_{ag \in Ag} t'_{ag} \wedge f_C \wedge f_\varphi$$

gdzie t'_{ag} jest formułą opisującą zbiór celów agenta ag powiększony o stany z bliskiego sąsiedztwa (z parametrem ε). Formuły f_C i f_φ rozważaliśmy w poprzednich problemach.

Przykład 5.3

Niech $t=1/3$ a $\varepsilon=1/2$. Przyjmijmy następującą funkcję odległości:

$$d(s, s') = \frac{1}{\text{card}(A_{ag})} \sum_{a \in A_{ag}} |s(a) - s'(a)|$$

Jeśli $d(s, s') < 1/2$ to stany s i s' są sobie bliskie. Dla agenta ag_1 nie można znaleźć żadnych stanów bliskich do tych z tablicy stanów lokalnych. Wszystkie stany bliskie do stanów z tablic stanów lokalnych agentów ag_2 i ag_3 przedstawia odpowiednio Tabela 14 i Tabela 15. W tablicach przedstawiono również ocenę nowych stanów lokalnych zakładając, że jest ona taka sama jak stanów bliskich, na których podstawie je otrzymaliśmy.

stany lokalne	s	t	u	e_{ag_1}
s_1	2	1	2	1
	2	2	2	
	2	1	1	
	1	1	2	
s_2 (bieżący)	2	2	1	1
	1	2	1	
	2	2	0	
s_3	2	1	1	2/3
	2	1	0	
	2	0	1	
	1	1	1	
s_4	1	2	1	1/3
	2	2	1	
	1	2	2	
	0	2	1	
	1	2	0	
s_5	1	1	2	1/3
	2	1	2	
	1	0	2	
s_6	1	1	1	0
	2	1	1	
	0	1	1	
	1	0	1	
	1	1	0	
s_7	2	0	1	0
	2	0	0	
	2	0	2	
s_8	0	1	2	0
	0	2	2	
	0	0	2	

Tabela 14 Stany lokalne agenta ag_2 z bliskim sąsiedztwem.

stany lokalne	k	l	m	e_{ag_3}
s_1	2	2	2	1
	2	1	2	
	2	2	1	
s_2	1	2	2	2/3
	1	1	2	
	0	2	2	
s_3	1	1	2	1/3
	2	1	2	
	1	2	2	
	0	1	2	
	1	0	2	
s_4	1	1	1	1/3
	2	1	1	
	1	2	1	
	0	1	1	
	1	0	1	
	1	1	0	
s_5 (bieżący)	2	0	1	0
	2	0	0	
	2	0	2	

Tabela 15 Stany lokalne agenta ag_3 z bliskim sąsiedztwem.

Formuła t'_{ag_1} jest taka sama jak t_{ag_1} . Sposób znalezienia formuł na podstawie tablic informacyjnych jest zawsze taki sam. Przekształcamy tablice stanów lokalnych z bliskim sąsiedztwem w tablice decyzyjne (decyzja 1 dla stanów o ocenie co najmniej 2/3). Następnie znajdujemy minimalne reguły decyzyjne dla decyzji 1, a lewe strony reguł tworzą poszukiwane formuły t'_{ag_2} i t'_{ag_3} . Odpowiednio mamy następujące formuły:

$$t'_{ag_1} = a_2 \vee b_2$$

$$t'_{ag_2} = s_2 t_1 u_2 \vee s_2 t_2 u_2 \vee s_1 t_1 u_2 \vee s_1 t_2 u_1 \vee s_2 t_2 u_0 \vee s_2 t_1 u_0 \vee s_1 t_1 u_1 \vee s_2 u_1$$

$$t'_{ag_3} = k_2 l_2 \vee k_0 l_2 \vee k_2 l_1 m_2 \vee k_1 l_2 m_2 \vee k_1 l_1 m_2$$

Po wykonaniu koniunkcji powyższych formuł z formułami f_C i f_ϕ otrzymujemy pięć propozycji rozwiązywania konfliktu (w tym nowa propozycja sytuacji niekonfliktowej – porównaj z punktem 5.2):

$$f' = a_1 b_2 s_2 t_1 u_1 k_2 l_2 m_1 \vee a_1 b_2 s_2 t_1 u_2 k_2 l_2 m_1 \vee a_1 b_2 s_2 t_1 u_1 k_2 l_2 m_2 \vee a_1 b_2 s_2 t_1 u_2 k_2 l_2 m_2 \vee a_1 b_2 s_1 t_2 u_1 k_2 l_2 m_2$$

5.3 Konsensus na poziomie globalnym (ocena agentów)

W tym punkcie będziemy poszukiwali konsensusu opierając się na preferencjach agentów.

DANE WEJŚCIOWE

System z ograniczeniami M_{Ag} zdefiniowany w rozdziale 3.

h – akceptowany poziom preferencji.

WYNIK

Zbiór Sp wszystkich sytuacji niekonfliktowych S' tj. $Sp = \{S': Cp_{Ag}^h(S') = 0\}$.

ALGORYTM

Najprostszy algorytm polega na wygenerowaniu wszystkich osiągalnych sytuacji i zastosowaniu do nich wartości globalnej funkcji preferencji. Zbiór $Spref_{Ag}(h)$ jest poszukiwanym zbiorem rozwiązań Sp . Wyznaczamy go przez przekształcenie tablicy sytuacji osiągalnych w tablicę decyzyjną (decyzja 1 dla $p(S) \geq h$) i wyznaczenie reguły decyzyjnej podobnie jak dla zbioru sytuacji dobrych $Score_{Ag}(t)$. Ze względu na ogromną liczbę sytuacji

osiągalnych w rzeczywistych konfliktach powyższy algorytm wydaje się być mało zachęcający.

Inny algorytm rozwiązania tego problemu polega na możliwie równomiernym podziale poziomu preferencji na wszystkich agentów biorących udział w konflikcie np. w następujący sposób (dla n agentów $1 \leq j \leq n$): $h \leq h_1 + \dots + h_j + \dots + h_n$. Załóżmy, że takich możliwych podziałów (z określonym poziomem różnicy między agentami) jest m . Następnie należy znaleźć opis zbioru celów każdego agenta z zadaniem poziomem h_{ij} (w i -tym podziale): $T_{ag_j}^i = \{s \in U_{ag} : e_{ag}(s) > h_{ij}\}$ – formuła $t_{ag_j}^i$. Koniunkcja komponentów $t_{ag_j}^i$ znalezionych w poprzednim kroku tworzy formułę boolowską f_i . Implikanty pierwsze formuły f_i zweryfikowanej z ograniczeniami są poszukiwanym rozwiązaniem problemu konsensusu w i -tym podziale. Zakładając m możliwych podziałów poziomu preferencji h dyzjunkcja $f = f_1 \vee \dots \vee f_m$ opisuje zdefiniowany powyżej problem konsensusu. Podsumowując:

$$f = \left(\bigvee_{1 \leq i \leq m} f_P^i \right) \wedge f_\varphi$$

gdzie m jest liczbą możliwych podziałów poziomu preferencji h , f_φ opisuje ograniczenia a formuła f_P^i jest formułą opisującą zbiór $S_{pref_{Ag}}(h_i)$ w i -tym podziale:

$$f_P^i = \bigwedge_{ag \in Ag} t_{ag}^i$$

gdzie t_{ag}^i opisuje zbiór celów agenta ag w i -tym podziale.

Przykład 5.4

Niech poziom preferencji dla rozważanego konfliktu wynosi $h=8/3$. Rozważymy tylko sensowne podziały poziomu preferencji tzn. takie, które dają niepuste zbiory celów dla wszystkich agentów (dopuszczalna różnica w podziale poziomu preferencji wynosi $1/3$):

podział 1: $h_{11} = 1, h_{12} = 1, h_{13} = 2/3$

podział 2: $h_{21} = 1, h_{22} = 2/3, h_{23} = 1$

podział 3: $h_{31} = 2/3, h_{32} = 1, h_{33} = 1$

Sposób znalezienia formuły opisującej zbiór celów przedstawiono w rozdziale 3.1.6. W ten sam sposób zostały wyznaczone formuły t_{agj}^i :

$$\begin{aligned} t_{ag1}^1 &= a_2 b_2, & t_{ag2}^1 &= s_2 t_2 \vee s_2 u_2, & t_{ag3}^1 &= l_2 \vee k_2 m_2 \\ t_{ag1}^2 &= a_2 b_2, & t_{ag2}^2 &= s_2 t_1 \vee s_2 t_2 \vee s_2 u_2, & t_{ag3}^2 &= k_2 l_2 \vee k_2 m_2 \\ t_{ag1}^3 &= a_2 \vee b_2, & t_{ag2}^3 &= s_2 t_2 \vee s_2 u_2, & t_{ag3}^3 &= k_2 l_2 \vee k_2 m_2 \end{aligned}$$

Formuła f jest następująca:

$$f = ((a_2 b_2 \wedge (s_2 t_2 \vee s_2 u_2) \wedge (l_2 \vee k_2 m_2)) \vee (a_2 b_2 \wedge (s_2 t_1 \vee s_2 t_2 \vee s_2 u_2) \wedge (k_2 l_2 \vee k_2 m_2)) \vee (a_2 \vee b_2) \wedge (s_2 t_2 \vee s_2 u_2) \wedge (k_2 l_2 \vee k_2 m_2)) \wedge f_\varphi$$

Po redukcji:

$$\begin{aligned} f &= a_1 b_2 s_2 t_1 u_2 k_2 l_2 m_1 \vee a_2 b_0 s_2 t_2 u_1 k_2 l_0 m_2 \vee a_2 b_0 s_2 t_2 u_2 k_2 l_0 m_2 \vee a_2 b_1 s_2 t_2 u_1 k_2 l_0 m_2 \vee a_2 b_1 s_2 t_2 u_2 k_2 l_0 m_2 \vee \\ & a_1 b_2 s_2 t_1 u_2 k_2 l_0 m_2 \vee a_1 b_2 s_2 t_1 u_2 k_2 l_1 m_2 \vee a_1 b_2 s_2 t_1 u_2 k_2 l_2 m_2 \vee a_2 b_2 s_2 t_2 u_1 k_2 l_0 m_2 \vee a_2 b_2 s_2 t_2 u_2 k_2 l_0 m_2 \end{aligned}$$

Biorąc pod uwagę tylko preferencje agentów można znaleźć 10 sytuacji bezkonfliktowych.

5.4 Konsensus na poziomie globalnym uwzględniający dobro wspólne i preferencje agentów

Poprzednio rozważany problem konsensusu rozszerzymy o wymaganie, aby osiągnięty konsensus uwzględniał dobro wspólne agentów.

DANE WEJŚCIOWE

System z ograniczeniami M_{Ag} zdefiniowany w rozdziale 3.

h – akceptowany poziom preferencji.

t – akceptowany poziom jakości sytuacji globalnych.

WYNIK

Zbiór Spg wszystkich sytuacji niekonfliktowych S' – takich, dla których $Cp_{Ag}^h(S') = 0$ i równocześnie $Cg_{Ag}^t(S') = 0$ dla $S' \in Spg$ tzn. $Spg = \{S': Cp_{Ag}^h(S') = 0 \text{ oraz } Cg_{Ag}^t(S') = 0\}$

ALGORYTM

Należy wyznaczyć zbiór $Spref_{Ag}(h)$ i zweryfikować go ze zbiorem $Score_{Ag}(t)$ oraz z ograniczeniami. Sposób wyznaczenia opisu zbioru $Spref_{Ag}(h)$ (formuła f_P) został przedstawiony w poprzednim punkcie 5.3, natomiast zbioru $Score_{Ag}(t)$ (formuła f_C) w punkcie 3.4.1.1. Implikanty pierwsze poniższej formuły f wyznaczają wszystkie sytuacje niekonfliktowe i opisują zbiór Spg .

$$f = f_P \wedge f_C \wedge f_\phi$$

W praktyce pierwszym krokiem algorytmu jest rozwiązanie poprzedniego problemu konsensusu (punkt 5.3). Następnie otrzymaną formułę należy przemnożyć (zredukować) z formułą f_C .

Przykład 5.5

W omawianym przykładowym konflikcie rozwiązanie wyznacza formuła:

$$f = ((a_2 b_2 \wedge (s_2 t_2 \vee s_2 u_2) \wedge (l_2 \vee k_2 m_2)) \vee (a_2 b_2 \wedge (s_2 t_1 \vee s_2 t_2 \vee s_2 u_2) \wedge (k_2 l_2 \vee k_2 m_2)) \vee (a_2 \vee b_2) \wedge (s_2 t_2 \vee s_2 u_2) \wedge (k_2 l_2 \vee k_2 m_2)) \wedge (a_1 k_2 l_2 m_2 \vee a_1 s_2 k_2 l_2 m_1 \vee b_2 s_2 k_2 l_2 m_1 \vee a_1 u_2 k_2 l_2 m_1 \vee b_2 u_2 k_2 l_2 m_1 \vee a_1 t_1 k_2 l_2 m_1 \vee b_2 t_1 k_2 l_2 m_1 \vee k_1 l_1 m_1 \vee k_1 l_2 m_2 \vee t_1 u_2 k_2 l_2 m_1 \vee s_1 u_2 k_2 l_2 m_1) \wedge f_\phi$$

Po redukcji okazuje się, że formuła f ma dwa implikanty pierwsze

$$f = a_1 b_2 s_2 t_1 u_2 k_2 l_2 m_1 \vee a_1 b_2 s_2 t_1 u_2 k_2 l_2 m_2$$

Można zauważyć, że otrzymany wynik zawiera się w rozwiązaniu problemu konsensusu na poziomie globalnym przy obiektywnej ocenie z uwzględnieniem lokalnych preferencji agentów (rozdział 5.2).

5.5 Algorytmy poszukiwania konsensusu

Wnioskowanie boolowskie zastosowane do rozwiązywania problemów konsensusu prowadzi do redukcji złożonej formuły boolowskiej – wyznaczenie implikantów pierwszych. Wielkość formuły (liczba komponentów) zależy przede wszystkim od liczby agentów biorących udział w konflikcie. Liczba dyzjunkcji w komponentach zależy w dużej mierze od liczby różnych atrybutów i ich wartości użytych do opisu stanowiska w danej sytuacji

konfliktowej. Pesymistyczny czas działania algorytmu redukcji formuł boolowskich jest wykładniczy w zależności od liczby komponentów i liczby dyzjunkcji w komponencie. Analiza sytuacji konfliktowych, w których bierze udział duża liczba agentów i które są opisane dużą liczbą atrybutów może powodować więc problemy pamięciowe i czasowe. Szczegółową dyskusję zastosowanych algorytmów można znaleźć w rozdziale 8.

ANALIZA ZMIENNOŚCI SYTUACJI KONFLIKTOWYCH

Sytuacja konfliktowa może ulegać zmianom, ze względu na ruchy wykonywane przez agentów (zmiany stanów bieżących). Każdy agent może wykonać ruch zmieniający jego stan posiadania zasobów. Może to wpływać na stany innych agentów (utrata pewnych zasobów) – mamy wtedy do czynienia z sytuacją, którą można interpretować jako użycie siły w sytuacji konfliktowej. Rozstrzygnięcie konfliktów z użyciem siły opiera się na pojęciu *mocy agenta*. Inne zmiany, jeśli ograniczenia na to pozwalają, mogą nie mieć wpływu na bieżące stany pozostałych agentów biorących udział w konflikcie. Niezadowolenie agentów z zajmowanego stanu (konflikt lokalny) wpływa na trwałość sytuacji. W tym rozdziale będziemy analizować sytuację konfliktową od strony agenta biorącego udział w konflikcie. Analiza ruchów agentów i poszukiwanie punktu stabilności jest charakterystyczne dla teorii gier [35]. W tym rozdziale przedstawimy również jak w ramach zaproponowanego w pracy modelu korzystamy z podstawowych pojęć teorii gier.

6.1 Zasoby

Opisując model wskazaliśmy skończoność pewnych, istotnych dla agentów, *zasobów* jako fundamentalny powód konfliktów (punkt 3.3). Jednym z takich zasobów w przykładowym konflikcie jest skończony zysk zakładu. Często jednak zasoby, o które toczy się konflikt trudno jest nawet nazwać. W zaproponowanym modelu wyspecyfikowanie zasobów jest nieistotne. Zasoby są opisane ograniczeniami pozwalającymi wyznaczyć zbiór sytuacji osiągalnych, a stany lokalne agentów odzwierciedlają stany posiadania zasobów przez poszczególnych agentów. Każdy ruch agenta (patrz 6.2) wiąże się z zajmowaniem lub zwalnianiem pewnych zasobów przez agentów (w naszym modelu nie można odróżnić czy dany ruch polega na zajęciu czy zwolnieniu zasobów). Dla danego agenta ograniczenia oraz stany innych agentów wyznaczają jego *pokojuwą przestrzeń ruchów*. Zajmowanie stanów z pokojowej przestrzeni ruchów można interpretować jako zajmowanie pewnych wolnych

zasobów lub zwalnianiem zasobów. Jeśli natomiast ruch agenta wymusza zmianę stanu innego agenta to mamy do czynienia z rozwiązaniami siłowymi – zagarnięcie zasobów siłą.

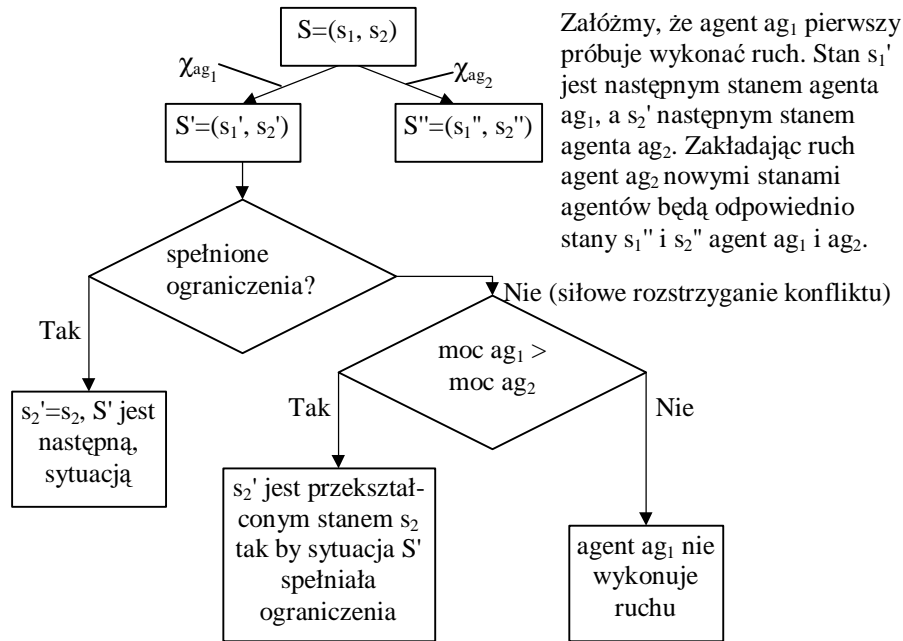
6.2 Ruchy agentów

Sytuacja konfliktowa zmienia się wraz ze zmianą stanów lokalnych agentów. Można rozważać rodzaj gry, gdy zmiany stanów nie podlegają globalnej kontroli i są niezależne (nie wynikają z ustaleń między agentami). Ze względu na ograniczenia, przejście agenta z jednego stanu do innego może wywoływać zmianę stanów innych agentów.

Dokładniej, *relacja przejścia* agenta ag jest relacją binarną χ_{ag} w $U_{ag} \times U_{ag}$. Przyjmujemy następującą interpretację. Jeśli s jest stanem bieżącym to jako możliwy stan następny agenta ag przyjmujemy dowolny stan s' taki, że $\langle s, s' \rangle \in \chi_{ag}$.

Relację χ_{ag} rozszerzamy na sytuacje. Ograniczymy się do przypadku dwóch agentów ag i ag' . Niech bieżąca sytuacja składa się ze stanów (s_1, s_2) pochodzących odpowiednio od agentów ag i ag' , a następna ze stanów (s_1', s_2') . Wtedy $(s_1, s_2)\chi_{ag}(s_1', s_2') \Leftrightarrow s_1\chi_{ag}s_1'$ oraz jeśli spełnione są ograniczenia (więzy zdefiniowane w danym konflikcie) to $s_2=s_2'$, w przeciwnym przypadku o ile $moc\ ag > moc\ ag'$ to s_2' powstaje przez przekształcenie s_2 tak, by (s_1', s_2') spełniało ograniczenia, w przeciwnym przypadku agent nie wykonuje ruchu. Tylko sytuacje spełniające ograniczenia (spełniające zależności opisujące ograniczenia) – nazywane sytuacjami osiągalnymi (punkt 3.3) mogą powstać w „świecie” opisanym z pomocą naszego modelu. Stan s_2' można wyznaczyć na podstawie stanu s_1' i ograniczeń z pomocą przestrzeni ruchów agenta ag' (patrz punkt 6.3 oraz 6.5.2). Rysunek 1 schematycznie przedstawia możliwość wykonania ruchu przez jednego z agentów. Moc agentów jest zdefiniowana w punkcie 6.5.1.

Wykonanie ruchu przez jednego z agentów może ograniczyć możliwość ruchu innego agenta. Dlatego ważna może być kolejność ruchów agentów. Kolejność może być niedeterministyczna, zgodna z rozkładem zmiennej losowej określającej częstotliwość ruchu każdego z agentów czy na przykład naprzemienna.



Rysunek 1 Schemat możliwości wykonania ruchu dla układu dwóch agentów biorących udział w konflikcie.

Z grą w teorii gier związane jest pojęcie funkcji wypłaty i zwycięzcy. W sytuacjach konfliktowych celem agenta jest zajęcie jak najlepszego stanu w *sytuacji trwale akceptowalnej (trwalej)*. Agent dąży do zajęcia stanu ze zbioru celów, przy czym jest ograniczony *przestrzenią rozwiązań*. Jeśli agenci dbają o dobro wspólne, dodatkowo ich możliwości są ograniczone przez zbiór stanów globalnie dobrych $Score_{Ag}(t)$. Możliwe jest, że sytuacja globalnie dobra nie jest *trwała*.

6.2.1 Koszt zmiany stanu

Z każdą zmianą stanu jest związany *koszt*, jaki dany agent musi ponieść, aby przejść do innego stanu. Koszt jest opisany funkcją postaci: $c: U_{ag} \times U_{ag} \rightarrow \mathbf{R}_+$. Przyjmijmy dla przykładu, że wartość kosztu zmiany stanu odpowiada wartości odległości stanu bieżącego od stanu docelowego tzn. $c(s, s') = d(s, s')$ gdzie $s, s' \in U_{ag}$, liczonej za pomocą funkcji (patrz punkt 3.1.4):

$$F(r_1, \dots, r_m) = \frac{1}{m} \sum_{1 \leq i \leq m} r_i$$

Problem kosztu związanego ze zmianą stanu może być również uwzględniony przy poszukiwaniu konsensu (rozdział 5). Wtedy dodatkowym celem przy poszukiwaniu konsensusu może być minimalizacja kosztu zmiany stanu dla wszystkich agentów. Jednak znaczenie kosztu zmiany stanu jest większe, gdy analizujemy ruchy agentów, drogę dochodzenia do konsensusu. Całkowity koszt uzyskania konsensusu, liczony jako suma kosztów kolejnych zmian stanu, może bardzo różnić się dla poszczególnych agentów w zależności od wybranej drogi dochodzenia do konsensusu.

6.2.2 Trwałość sytuacji

Agenci są w *sytuacji trwałej*, jeśli w tej sytuacji są spełnione ich preferencje (są zadowoleni z sytuacji). Ponieważ oceny stanów lokalnych mogą być ze sobą nieporównywalne wprowadzamy dwuwartościową *funkcję zadowolenia* agentów.

$$g_{ag}(s) = \begin{cases} 1 & \text{gdy } s \in T(ag) \\ 0 & \text{w przeciwnym razie} \end{cases}$$

tzn. agent ag jest zadowolony gdy osiągnął stan ze zbioru celów.

Współczynnik trwałości sytuacji S będzie liczony jako:

$$tb(S) = \frac{\sum_{ag \in Ag} g_{ag}(s_{ag})}{card(Ag)}$$

gdzie s_{ag} jest stanem agenta ag w sytuacji S .

Sytuację S nazywamy *h-trwała*, jeśli $tb(S) > h$, gdzie h jest zadany poziomem trwałości. Sytuacja S jest *trwała*, gdy $tb(S) = 1$ i *nietrwała*, gdy $tb(S) = 0$.

Można uznać, że sytuacja jest nietrwała, gdy zbyt wielu agentów jest w konflikcie lokalnym.

6.2.3 Funkcja wypłaty, zwycięzca

Zwycięzcą jest agent, który w sytuacji trwalej zajmuje stan preferowany (należący do jego zbioru celów). *Przestrzeń rozwiązań* dla danego agenta wyznacza zbiór sytuacji, w których agent będzie zwycięzcą.

Miarą zadowolenia agenta jest subiektywna ocena stanu lokalnego, w którym się znalazł. Dlatego *funkcja wypłaty* w naszym modelu bezpośrednio odpowiada funkcji celu (rozdział 3.1.5). W przeciwieństwie jednak do teorii gier *funkcja wypłaty* jest w naszym modelu modelowana za pomocą zbioru reguł decyzyjnych, co ma znaczenie w przypadku złożonych systemów negocjacji, gdy wiedza o systemach nie pozwala na skonstruowanie precyzyjnych modeli analitycznych.

6.3 Pokojowa przestrzeń ruchów

Jeśli agenci odrzucają rozwiązania siłowe (patrz Rysunek 1) to *przestrzeń ruchów* każdego agenta jest ograniczona przez stany innych agentów i ograniczenia. Dodatkowo zbiór sytuacji dobrych ogranicza *przestrzeń ruchów*, jeśli agenci biorą pod uwagę dobro wspólne. *Pokojowa przestrzeń ruchów* N_{ag} agenta ag w bieżącej sytuacji S jest zbiorem jego stanów lokalnych niesprzecznych z formułą:

$$v_{ag} = \left(\bigwedge_{\substack{ag' \in Ag \\ ag' \neq ag}} f_{ag',s} \right) \wedge f_C \wedge f_\varphi$$

gdzie $f_{ag',s}$ jest formułą opisującą bieżący stan lokalny s agenta ag' w S , f_C jest formułą opisującą zbiór sytuacji dobrych $Score(t)$ a f_φ opisuje ograniczenia. Implikanty pierwsze tej formuły wyznaczają wszystkie osiągalne sytuacje (dokładniej sytuacje niesprzeczne z tymi implikantami), do których agent ag może przejść z bieżącego stanu. Formuły f_C i f_φ zostały opisane odpowiednio w punktach 3.4.1.1 i 3.3.1. Z formuły v_{ag} wynika, że to ograniczenia i zbiór sytuacji dobrych wyznaczają *przestrzeń ruchów* agenta ag (formuła $\bigwedge_{\substack{ag' \in Ag \\ ag' \neq ag}} f_{ag',s}$ opisuje

bieżącą sytuację dla podzbioru agentów $Ag - \{ag\}$, która z założenia jest niesprzeczna z ograniczeniami). Jeśli *przestrzeń ruchów* jest pusta to przejście do innego stanu wymaga od agenta negocjacji z innymi agentami (zwolnienia przez nich zasobów) lub wymuszenie zmiany w sposób siłowy – ten drugi sposób rozważamy w punkcie 6.5. Jeżeli natomiast bierzemy pod uwagę ruchy, które agenci mogą wykonać np. prowadząc negocjacje to

potencjalna przestrzeń ruchów agenta ag jest zbiorem jego stanów lokalnych niesprzecznych z formułą:

$$vp_{ag} = \left(\bigwedge_{\substack{ag' \in Ag \\ ag' \neq ag}} t_{ag'} \right) \wedge f_C \wedge f_\varphi$$

gdzie $t_{ag'}$ jest formułą opisującą zbiór celów agenta ag' . Nie staramy się tutaj śledzić kolejnych kroków negocjacji (przechodzenie przez kolejne sytuacje) lecz zakładamy, że strony dążą do uzyskania stanów ze zbioru celów. Oczywiście strony mogą nie dojść do porozumienia (sprzeczne zbiory celów z ograniczeniami), dlatego formuła vp_{ag} może również być sprzeczna.

6.4 Przestrzeń rozwiązań

W kolejnych ruchach agenci dążą do uzyskania celu (stanu ze zbioru celów) w sytuacji trwałej. Zbiór sytuacji trwałych, w których agent ag jest zwycięzcą nazwiemy jego *przestrzenią rozwiązań* (agenta ag). Jeżeli agenci odrzucają rozwiązania siłowe to (*pokojowa*) *przestrzeń rozwiązań* jest ograniczona przez preferencje (zbiory celów) innych agentów i ograniczenia. Dodatkowo zbiór stanów dobrych ogranicza tę przestrzeń, jeśli agenci biorą pod uwagę dobro wspólne.

Na podstawie definicji trwałości możemy wyznaczyć liczbę agentów, których zbiory celów musimy wziąć pod uwagę. Niech $Ag'_h \subset Ag$ będzie dowolnym podzbiorem agentów bez konfliktu lokalnego, przy zadanym poziomie trwałości h , wtedy:

$$card(Ag'_h) > \lfloor h \cdot card(Ag) \rfloor$$

Przyjmując $ag \in Ag'_h$ zakładamy, że agent ag ma być zwycięzcą. Ogólnie przestrzeń rozwiązań agenta ag można przedstawić za pomocą następującej formuły boolowskiej:

$$r_{ag} = \left(\bigvee_{\substack{Ag'_h \subset Ag \\ ag \in Ag'_h}} \bigwedge_{ag' \in Ag'_h} t_{ag'} \right) \wedge f_C \wedge f_\varphi$$

gdzie $t_{ag'}$ jest formułą opisującą zbiór celów agenta ag' , f_C jest formułą opisującą zbiór sytuacji dobrych $Score(t)$ a f_φ opisuje ograniczenia. Jeśli formuła jest sprzeczna (patrz Uwaga

5) to nie jest możliwe znalezienie konsensusu dla agentów z Ag' w sytuacji opisanej przez: preferencje agentów (zbiory celów), wymaganie dobra wspólnego (formuła opisująca sytuacje dobre f_c) i ograniczenia – formuła f_ϕ (porównaj 5.2). Niestety formuła r_{ag} ma wykładniczy rozmiar w zależności do liczby agentów. Dokładniej jeżeli n oznacza liczbę wszystkich agentów a m liczbę agentów należących do Ag'_h , to formuła będzie składała się z $\binom{n}{m-1}$ dyzjunkcji, warunek $m \geq 1$ wynika z założenia, że rozważany agent ma być zwycięzcą. W przypadku dużej liczby agentów trzeba więc poszukiwać algorytmów heurystycznych. Zauważmy, że w rozważanym problemie najbardziej złożoną częścią jest wyznaczenie wszystkich takich podzbiorów Ag'_h , w których są spełnione preferencje rozważanego agenta. Uogólniając, jeżeli udałoby się nam wyznaczyć możliwie duży zbiór agentów zadowolonych w konflikcie (takich, dla których są spełnione preferencje), to mielibyśmy przybliżone rozwiązanie naszego problemu. Jeżeli rozważany agent należy do tego zbioru to łatwo wyznaczyć formułę r_{ag} , jeżeli nie należy to traktujemy formułę jako sprzeczną. Osobnym problemem jest wyznaczenie jak największego zbioru agentów zadowolonych w danym konflikcie. Jednym ze sposobów może być próba tworzenia takiego zbioru z agentów najbliższych sobie pod względem poglądów na dane historyczne (patrz koalicje – punkt 6.5.5). Zaczynając od najbliższych sobie agentów (w zaproponowanej mierze odległości) dołączamy kolejnych agentów sprawdzając za każdym razem spełnienie preferencji (przez wykonanie koniunkcji utworzonej formuły z formułą opisującą cele dołączanego agenta). Oczywiście ten sposób nie gwarantuje wyznaczenia największego zbioru agentów zadowolonych, ale liczba dyzjunkcji w tym algorytmie jest liniowo zależna od liczby agentów.

6.4.1 Problem zwycięzcy

Należy sprawdzić czy w bieżącej sytuacji dany agent może zostać zwycięzcą w sposób pokojowy. Ponadto należy wyznaczyć stan (stany), w którym zwycięstwo wiąże się z najmniejszym kosztem dla danego agenta.

DANE WEJŚCIOWE

System z ograniczeniami M_{Ag} zdefiniowany w rozdziale 3.

ag – rozpatrywany agent.

h – zadany poziom trwałości.

WYNIK

Zbiór wszystkich sytuacji, w których agent ag jest zwycięzcą.

ALGORYTM

Należy wyznaczyć przestrzeń rozwiązań agenta ag i zweryfikować ją z jego przestrzenią ruchów. Problem można więc przedstawić z pomocą formuły:

$$f_{ag} = v_{ag} \wedge r_{ag}$$

Formuły v_{ag} i r_{ag} zostały omówione odpowiednio w rozdziałach 6.3 i 6.4. Pesymistyczny czas działania algorytmu jest wykładniczy w zależności od wielkości formuł (patrz rozdział 8). Z kolei formuła r_{ag} jest wykładniczo zależna od liczby agentów. Łatwiej więc opisać ten problem niż go rozwiązać dla dużej liczby agentów. Jak wspomniano w poprzednim punkcie (6.4) niezbędne jest opracowanie algorytmów heurystycznych dla praktycznego rozwiązania problemu.

Obliczając wartości kosztu dla każdego stanu agenta ag należącego do rozwiązania problemu zwycięzcy, znajdujemy stan (stany) o minimalnym koszcie związanym ze zmianą stanu.

Przykład 6.1

Niech zadany poziom trwałości w rozważanym konflikcie między pracodawcą i pracownikami wynosi $1/3$. W bieżącej sytuacji S jedynie agent ag_3 jest w konflikcie lokalnym dlatego współczynnik trwałości $tb(S) = 1/3 * (0+0+1) = 1/3$.

Aby sprawdzić, czy agent ag_3 może być zwycięzcą należy wyznaczyć jego przestrzeń rozwiązań i przestrzeń ruchów.

Ponieważ założyliśmy poziom trwałości $h=1/3$ to co najmniej dwóch agentów powinno zajmować stany z ich zbiorów celów ($card(Ag'_{\frac{1}{3}}) > \left\lfloor \frac{1}{3} \cdot 3 \right\rfloor$).

Formuła opisująca przestrzeń rozwiązań dla agenta ag_3 jest następująca:

$$r_{ag_3} = (t_{ag_3} \wedge t_{ag_1} \vee t_{ag_3} \wedge t_{ag_2}) \wedge f_C \wedge f_\varphi$$

$$r_{ag_3} = a_1b_0s_2t_1u_1k_2l_2m_1 \vee a_1b_0s_2t_1u_2k_2l_2m_1 \vee a_1b_1s_2t_1u_1k_2l_2m_1 \vee a_1b_1s_2t_1u_2k_2l_2m_1 \vee a_1b_2s_2t_1u_1k_2l_2m_1 \vee a_1b_2s_2t_1u_2k_2l_2m_1 \vee a_1b_0s_2t_1u_1k_2l_2m_2 \vee a_1b_0s_2t_1u_2k_2l_2m_2 \vee a_1b_1s_2t_1u_1k_2l_2m_2 \vee a_1b_1s_2t_1u_2k_2l_2m_2 \vee a_1b_2s_2t_1u_1k_2l_2m_2 \vee a_1b_2s_2t_1u_2k_2l_2m_2$$

Sytuacje opisane za pomocą formuły r_{ag_3} są sytuacjami trwałymi z parametrem h i definiują przestrzeń rozwiązań dla agenta ag_3 (w tych stanach ag_3 będzie zwycięzcą).

Przestrzeń ruchów wyznaczymy za pomocą następującej formuły:

$$v_{ag_3} = (a_2b_1 \wedge s_2t_2u_1) \wedge f_C \wedge f_\varphi$$

Stany a_2b_1 i $s_2t_2u_1$ są stanami bieżącymi odpowiednio agentów ag_1 i ag_2 (patrz punkt 6.3). Przestrzeń ruchów agenta ag_3 jest pusta, bo $v_{ag_3} \wedge r_{ag_3}$ jest formułą sprzeczną. Agent ag_3 nie może być zwycięzcą w bieżącej sytuacji bez jej rekonstrukcji.

Natomiast, gdy nie bierzemy pod uwagę dobra wspólnego (f_C) agent ag_3 może być zwycięzcą w jednej sytuacji (którą można osiągnąć na drodze pokojowej):

$$f_{ag_3} = a_2b_1s_2t_2u_1k_2l_0m_2$$

Wartość kosztu zmiany stanu ze stanu bieżącego do stanu opisanego przez f_{ag_3} dla agenta ag_3 wynosi: $1/3(|2-2|+|0-0|+|2-1|)=1/3$.

6.5 Rozwiązania siłowe

Pokojowa przestrzeń ruchów wynika z zajmowanych przez agentów stanów i ograniczeń. Ograniczenia (patrz 3.3) pochodzą ze skończoności zasobów a bieżące stany lokalne odzwierciedlają zajmowane zasoby przez poszczególnych agentów. W poprzednich rozdziałach zakładaliśmy, że agenci dążą do konsensusu w sposób pokojowy. Dlatego, zajęcie pewnych zasobów w danej sytuacji musiało wiązać się ze zwolnieniem ich przez innego agenta (albo zasoby były wolne – niczyje). Zwolnienie zasobu polega na wykonaniu takiego ruchu przez pewnego agenta (lub podzbiór agentów), że ruch innego agenta staje się możliwy do wykonania (ze względu na ograniczenia).

W przypadku, gdy agenci dopuszczają rozwiązania siłowe każdy agent może zająć dowolny stan ze zbioru sytuacji osiągalnych (porównaj punkt 6.1). W tym rozdziale będziemy rozważać rozwiązania siłowe konfliktów [40].

6.5.1 Moc agentów

Agenci dysponują różną siłą (znaczeniem, liczebnością armii itd.) wyznaczoną w oparciu o dane historyczne. (Jeżeli brak danych historycznych moc agentów może być podana przez eksperta). Pomocne przy analizie konfliktu jest założenie znajomości *mocy* agentów w danej sytuacji, wyrażonej np. z pomocą funkcji $w: Ag \rightarrow [0, 1]$, która każdemu agentowi biorącemu udział w konflikcie przypisuje wartość jego mocy (siły) w ten sposób, że:

$$\sum_{ag \in Ag} w_{ag} = 1$$

Wartość mocy może odpowiadać wadze (znaczeniu) danego agenta (patrz Uwaga 9).

Przykład 6.2

Zalóżmy następujące wartości mocy agentów dla naszego konfliktu (Tabela 16).

agent	w_{ag}
ag_1	$2/7$
ag_2	$2/7$
ag_3	$3/7$

Tabela 16 Moc agentów w konflikcie między pracodawcą i pracownikami

6.5.2 Podział łupów

W wypadku rozwiązań siłowych konfliktów agenci o większej mocy mają pierwszeństwo w zajmowaniu stanów (zasobów) przed agentami o mniejszej mocy. Odwrotnie, agenci o mniejsze mocy są ograniczeni przy zmianie stanu zgodnie z relacją przejścia, przez stany zajmowane przez agentów o większej mocy. Proces ten będziemy nazywali *podziałem łupów*.

Jeżeli agenci dysponują tą samą mocą mogą wzajemnie zajmować (zabierać) sobie zasoby. Dlatego jeżeli nie istnieje rozwiązanie trwale, agenci muszą się porozumieć – znaleźć konsensus na drodze pokojowej.

W rozwiązaniach siłowych przestrzeń ruchów agenta ag jest ograniczona przez stany lokalne agentów o większej mocy, ograniczenia i ewentualnie sytuacje dobre, jeśli agenci dbają o dobro wspólne. Niech $Ag' \subset Ag$ będzie zbiorem agentów o większej mocy niż posiada agent ag tzn.:

$$Ag' = \{ag' : w_{ag'} > w_{ag}, ag \neq ag' \in Ag\}$$

Przestrzeń ruchów agenta ag będzie wtedy opisana formułą:

$$v_{ag} = \bigwedge_{\substack{ag' \in Ag' \\ ag' \neq ag}} f_{s(ag')} \wedge f_C \wedge f_\varphi$$

gdzie $f_{s(ag')}$ jest formułą opisującą bieżący stan lokalny s agenta ag' , f_C jest formułą opisującą zbiór sytuacji dobrych $Score(t)$ a f_φ opisuje ograniczenia.

6.5.3 Problem zwycięstwa przy podziale łupów

Należy sprawdzić czy agent ag dysponujący mocą w_{ag} może zostać zwycięzcą w konflikcie, w którym agenci używają siły.

DANE WEJŚCIOWE

System z ograniczeniami M_{Ag} zdefiniowany w rozdziale 3,

ag – rozpatrywany agent,

h – zadany poziom trwałości,

w – funkcja mocy agentów biorących udział w konflikcie.

WYNIK

Zbiór wszystkich sytuacji, w których agent ag jest zwycięzcą.

ALGORYTM

Zwycięski dla agenta ag podział łupów uzyskamy weryfikując jego przestrzeń ruchów ze zbiorem celów. Następnie należy odrzucić sytuacje, dla których nie jest spełniony warunek trwałości $tb(S) > h$. Rozważany problem opisuje następująca formuła:

$$l_{ag} = t_{ag} \wedge v_{ag}$$

gdzie t_{ag} jest formułą opisującą zbiór celów agenta ag a v_{ag} jego przestrzeń ruchów.

Przykład 6.3

Przyjmijmy poziom trwałości $h=1/3$ oraz wartości mocy agentów dla rozważanego konfliktu – Tabela 16. Załóżmy ponadto, że agent ag_3 (dysponujący największą mocą) zajmie pierwszy stan ze zbioru celów (opisany formułą $f_{s(ag_3)} = k_2 l_2 m_2$).

Zwycięski podział łupów dla agenta ag_1 opisuje formuła:

$$l_{ag_1} = (a_2 \vee b_2) \wedge k_2 l_2 m_2 \wedge f_c \wedge f_\phi$$

Po redukcji:

$$l_{ag_1} = a_1 b_2 s_2 t_1 u_1 k_2 l_2 m_2 \vee a_1 b_2 s_2 t_1 u_2 k_2 l_2 m_2 \vee a_1 b_2 s_1 t_2 u_1 k_2 l_2 m_2 \vee a_1 b_2 s_1 t_2 u_2 k_2 l_2 m_2$$

Sprawdźmy czy w sytuacjach opisanych przez l_{ag_1} również agent ag_2 może być zwycięzcą. W tym celu znajdziemy koniunkcję $t_{ag_2} \wedge l_{ag_1}$, której implikanty pierwsze opisują sytuacje, w których wszyscy agenci są zwycięzcami.

$$t_{ag_2} \wedge l_{ag_1} = a_1 b_2 s_2 t_1 u_1 k_2 l_2 m_2 \vee a_1 b_2 s_2 t_1 u_2 k_2 l_2 m_2$$

Można zauważyć, że przy podziale łupów:

- agent o największej mocy zawsze jest zwycięzcą,
- wybór stanu przez agenta o większej mocy ma wpływ (zgodnie z więzami) na to czy agenci o mniejszej mocy będą zwycięzcami.

6.5.4 Rozwiązania siłowe a sytuacje trwałe

Jak już wspomniano wyżej sytuacja trwała jest rozwiązaniem konfliktu. W wyniku „bezmyślnego” podziału łupów agenci mogą znaleźć się w sytuacji nietrwalej. Mimo więc zaspokojenia własnych preferencji agenci nie osiągną zwycięstwa (trwałego pokoju – konsensusu).

Co więcej każdy agent broni zwycięstwa z siłą równą jego mocy. Dlatego tylko silny agent może dokonać rozłamu i doprowadzić do nowego podziału łupów.

6.5.5 Koalicje

Aby zwiększyć swoje możliwości przy podziale łupów agenci tworzą koalicje. Koalicje są formalnie podzbiorami agentów. Koalicje mają następujące właściwości:

- moc koalicji jest sumą mocy agentów do niej należących,
- konsensus w koalicji jest znajdowany w sposób pokojowy.

Podstawą wyznaczania koalicji może być analiza subiektywnej oceny agentów różnych sytuacji. Można założyć, że jeżeli ocena agentów dla wielu różnych sytuacji jest zbieżna to agenci mogą stworzyć koalicję. Przy tym założeniu koalicję dwóch agentów można zdefiniować jako relację tolerancji ρ taką, że:

$$\langle ag, ag' \rangle \in \rho \Leftrightarrow D(ag, ag') \leq t$$

gdzie t jest poziomem zbieżności poglądów koalicjantów, a D jest funkcją odległości zdefiniowaną w następujący sposób:

$$D(ag, ag') = F(f(e_{ag}(S_1), e_{ag'}(S_1)), \dots, f(e_{ag}(S_m), e_{ag'}(S_m)))$$

$e_{ag}(S)$ jest subiektywną oceną stanu agenta ag w sytuacji S , a funkcje f i F są odpowiednio dobranymi funkcjami np.:

$$D(ag, ag') = \frac{1}{m} \sum_{i=1}^m |e_{ag}(S_i) - e_{ag'}(S_i)|$$

Koalicja C jest zbiorem agentów $C \subseteq Ag$ takim, że: $\forall (ag, ag' \in C) \max D(ag, ag') \leq t$.

Przykład 6.4

Rozważmy oceny agentów dla następujących sytuacji (Tabela 17) i wynikające z nich koalicje.

Sytuacje	<i>a</i>	<i>b</i>	<i>s</i>	<i>t</i>	<i>u</i>	<i>k</i>	<i>l</i>	<i>m</i>	e_{ag_1}	e_{ag_2}	e_{ag_3}
S_1	1	2	1	2	1	2	0	1	2/3	1/3	0
S_2	1	2	1	1	1	2	2	1	2/3	0	1/3
S_3	1	2	1	2	1	1	2	1	2/3	1/3	1/3
S_4	1	2	1	2	1	2	2	1	2/3	1/3	1/3
S_5	1	2	2	2	2	1	2	1	2/3	1	1/3
S_6	1	1	1	1	1	2	2	0	0	0	2/3
S_7	1	0	1	1	1	0	2	1	0	0	1/3
S_8	1	1	2	1	1	1	0	1	0	2/3	0
S_9	1	2	2	2	1	2	0	1	2/3	1	0
S_{10}	1	2	1	2	1	1	0	2	2/3	1/3	0
S_{11}	1	2	1	1	2	0	2	1	2/3	1/3	1/3
S_{12}	2	1	2	2	2	2	2	1	2/3	1	1/3

Tabela 17 Subiektywna ocena różnych sytuacji.

Odległość (zbieżność poglądów) między agentami obliczamy zgodnie z funkcją $D(ag, ag')$ np.:

$$D(ag_1, ag_2) = 1/12(|2/3-1/3|+|2/3-0|+|2/3-1/3|+|2/3-1/3|+|2/3-1|+|0-0|+|0-0|+|0-2/3|+|2/3-1|+|2/3-1/3|+|2/3-1/3|+|2/3-1/3|) = 1/3$$

Zbieżność poglądów agentów można przedstawić w postaci tablicy odległości (Tabela 18).

agenci	ag_1	ag_2	ag_3
ag_1			
ag_2	0,33		
ag_3	0,40	0,49	

Tabela 18 Tablica odległości - zbieżności poglądów agentów.

Zakładając wymagany poziom zbieżności poglądów agentów na $t=0,35$ można wyznaczyć koalicję składającą się z dwóch agentów ag_1 i ag_2 .

Zalóźmy więc, że agenci ag_1 i ag_2 utworzą koalicję i zajmą stany z ich zbiorów celów odpowiednio opisane formułami $f_{s(ag_1)}=a_2b_2$ i $f_{s(ag_2)}=s_2t_2u_1$. Przestrzeń ruchów agenta ag_3 będzie opisana formułą.

$$v_{ag_3}=a_2b_2 \wedge s_2t_2u_1 \wedge f_\emptyset$$

Gdy nie bierzemy pod uwagę dobra wspólnego, to przestrzeń ruchów agenta ag_3 będzie zawierała tylko jedną sytuację:

$$v_{ag_3}= a_2b_2s_2t_2u_1k_2l_0m_2$$

Stworzenie koalicji przez agentów ag_1 i ag_2 ma wyraźny wpływ na podział łupów (porównaj Przykład 6.3).

ANALIZA STRATEGII

Zmiana stanu lokalnego danego agenta jest związana z zajęciem pewnych zasobów będących w posiadaniu innych agentów lub wolnych, albo z ich zwolnieniem (patrz punkt 6.1) – zawsze powoduje zmianę sytuacji. Ruchy agentów mogą wynikać z uzgodnień między agentami (wynik negocjacji) lub mogą być wykonywane wedle zaplanowanej strategii (jeśli taką dany agent posiada). Analiza strategii agentów pozwala na monitorowanie sytuacji i ostrzeżenie przed zbliżającym się konfliktem. Możliwe jest także wskazanie sposobu rekonstrukcji sytuacji konfliktowej do sytuacji bezpiecznej z uwzględnieniem strategii.

7.1 Definicja strategii

Strategia agenta opisuje jego możliwe ruchy w zadanej sytuacji określonej przez stany lokalne innych agentów i ograniczenia. Bardziej dokładnie *strategia agenta* jest relacją określoną na zbiorze sytuacji. Strategia agenta jest przemyślaną taktyką zmiany stanów. W przyjętym modelu uzyskuje się ją analizując dane historyczne – opisujące zmiany sytuacji w wyniku ruchu agenta w przeszłości. Ponieważ w rzeczywistych konfliktach nie możemy liczyć na pełny zbiór wszystkich możliwych odpowiedzi (posunięć) agenta na wszystkie zaistniałe sytuacje aproksymujemy strategię znajdując ich opis w postaci reguł decyzyjnych z posiadanych danych w oparciu o wnioskowanie indukcyjne [29]. Ten sposób postępowania, stosowany w całej pracy (stany lokalne, sytuacje dobre) istotnie wyróżnia zaproponowany model w porównaniu z modelami opartymi o metody analityczne czy statystyczne [33], [48].

Przykład 7.1

Niech będzie dana tablica z historycznymi danymi, opisująca przejścia z sytuacji 1 do sytuacji 2 w konflikcie z udziałem agentów ag_1 , ag_2 i ag_3 .

sytuacja 1								sytuacja 2							
a	b	s	t	u	k	l	m	a	b	s	t	u	k	l	m
1	1	2	1	1	2	2	0	1	0	2	1	1	2	2	1
1	1	2	1	1	2	0	1	1	1	1	2	1	2	2	1
1	2	1	2	1	2	0	1	1	0	2	1	2	2	1	2
1	0	1	2	2	2	2	2	1	1	2	1	2	2	2	2
2	1	2	2	2	2	0	2	1	2	1	2	2	2	0	2
1	0	2	1	1	2	2	1	1	1	2	1	1	2	2	0
1	1	1	2	1	2	2	1	1	1	2	1	1	2	0	1
1	0	2	1	2	2	1	2	1	2	1	2	1	2	0	1
1	1	2	1	2	2	2	2	1	0	1	2	2	2	2	2
1	2	1	2	2	2	0	2	2	1	2	2	2	2	0	2
1	2	1	2	2	2	2	2	2	2	2	2	1	2	0	2
1	0	2	1	2	2	2	2	1	0	1	2	1	2	2	2
1	0	2	1	1	2	1	0	1	0	1	2	1	2	2	0
2	2	2	2	1	2	0	2	1	2	1	2	2	2	2	2
1	0	1	2	1	2	2	2	1	0	2	1	2	2	2	2
1	0	1	2	1	2	2	0	1	0	2	1	1	2	1	0

Tabela 19 Tablica przejścia (dane historyczne)

Znajdujemy reguły decyzyjne traktując atrybuty z "sytuacji 2" jako decyzyjne i biorąc za każdym razem inny atrybut jako decyzję.

$$b_1 \vee b_0 \vee s_2 \vee t_1 \vee u_1 \vee m_0 \vee m_1 \vee a_2 \vee l_1 \rightarrow a_1$$

$$b_2u_2 \vee a_1u_2l_0 \vee s_1u_2l_0 \vee a_1b_2m_2 \vee b_2s_1m_2 \vee a_1l_0m_2 \vee s_1l_0m_2 \vee b_2l_2 \rightarrow a_2$$

$$b_1s_2l_2 \vee b_1t_1l_2 \vee m_0 \vee b_1u_2l_2 \vee b_2m_1 \vee a_1b_2u_1 \vee a_1b_1u_2 \vee b_2s_1u_1 \vee b_0s_1u_1 \vee s_1u_1l_0 \vee s_2u_2l_2 \vee s_1l_0m_1 \vee s_2l_2m_2 \vee a_1t_2u_1l_0 \vee t_2l_0m_1 \vee t_1l_2m_2 \vee b_1t_1u_2 \vee b_0t_2u_1 \vee b_1t_1m_2 \vee b_1l_2m_2 \vee u_1l_1 \vee t_1u_2l_2 \vee a_1b_1m_2 \vee b_0u_1m_2 \vee s_1u_1m_2 \vee u_1l_2m_2 \vee a_1u_1m_2 \rightarrow b_0$$

$$b_1t_2l_2 \vee a_1b_1l_0 \vee b_1u_1l_0 \vee b_2u_2l_0 \vee a_1s_2l_0 \vee t_1l_0 \vee b_1m_1 \vee b_0m_1 \vee s_2m_1 \vee t_1m_1 \vee b_0s_1u_2 \vee s_1u_2l_0 \vee s_1l_0m_2 \vee b_0t_2u_2 \vee b_1t_2u_1 \vee b_0s_2u_1l_2 \vee b_0t_1u_1l_2 \vee l_2m_1 \vee b_1s_1 \vee a_1b_1t_2 \vee a_1u_2l_0 \vee a_1l_0m_2 \rightarrow b_1$$

$$b_1t_2l_0 \vee b_1u_2l_0 \vee s_2u_2l_0 \vee s_2l_0m_2 \vee b_1t_2u_2 \vee a_2 \vee s_2t_2 \vee b_1t_2m_2 \vee b_1l_0m_2 \vee l_1m_2 \vee b_2s_2 \vee u_2l_1 \vee b_2l_2 \vee b_2u_1m_2 \vee s_2u_1m_2 \vee u_1l_0m_2 \rightarrow b_2$$

$$b_1m_2 \vee b_1l_0 \vee s_2l_0 \vee t_1l_0 \vee b_1s_2m_1 \vee b_0s_2m_0 \vee b_1t_1m_1 \vee b_0t_1m_0 \vee t_1m_2 \vee a_2 \vee s_2t_2 \vee b_1u_2 \vee s_2u_2 \vee s_2m_2 \vee t_1u_2 \vee l_1 \vee b_2s_2 \vee b_2u_1m_2 \vee u_1l_0m_2 \rightarrow s_1$$

$$u_1l_2 \vee l_2m_0 \vee l_2m_1 \vee b_1m_0 \vee b_2m_1 \vee b_0m_1 \vee b_2l_2 \vee t_2l_2 \vee a_1b_2 \vee s_1 \vee a_1t_2 \vee t_2m_1 \vee t_2m_0 \vee b_0t_2 \vee b_2u_2 \vee b_1t_2u_1 \vee a_1u_2l_0 \vee a_1l_0m_2 \vee b_0u_1m_2 \vee a_1u_1m_2 \rightarrow s_2$$

itd.

Reguły te oznaczają np. że agent ag_I może zgodnie ze strategią znaleźć się w stanie gdzie $a=1$ wtedy gdy np. $b=1$ lub $b=0$ lub $s=2$ itd.

7.2 Strategiczna przestrzeń sytuacji

Strategie wprowadzają nowe ograniczenia na przestrzeń ruchów agentów. W danej sytuacji możemy na podstawie strategii wyznaczyć wszystkie *dostępne sytuacje* zgodnie z tą strategią.

Wyznaczenie wszystkich możliwych dostępnych sytuacji z danej sytuacji polega na sprawdzeniu, które reguły strategii są dla tej sytuacji spełnione. W tym celu konstruujemy koniunkcję opisu danej sytuacji z lewymi stronami reguł (strategii). Jeśli koniunkcja jest niesprzeczna (patrz Uwaga 5) to prawa strona wyznacza wartość danego atrybutu w nowej sytuacji. Otrzymane sytuacje należy zweryfikować z ograniczeniami (więzami). Postępowanie to można przedstawić w następujący sposób.

Niech $fst(a_i)$ będzie opisem wyznaczonym przez lewą stronę reguły dla decyzji a_i , a f_s jest formułą opisującą rozpatrywaną sytuację.

$$f(a_i) = f_s \wedge fst(a_i)$$

Jeżeli $f(a_i) \equiv f_s$ to sytuacja S może zmienić się w taką sytuację S' , w której występuje a_i . Koniunkcja a_i dla $a \in A$ opisuje sytuację S' . Zbiór sytuacji S' jest opisany następującą formułą $f_Z(S)$.

$$f_Z(S) = \bigwedge_{\substack{a \in A \\ f(a_i) \equiv f_s}} (\bigvee_{i \in V_a} a_i) \wedge f_\phi$$

gdzie f_ϕ opisuje ograniczenia. Formuła f_Z opisuje *strategiczną przestrzeń sytuacji* osiągalnych z S . Zbiór sytuacji wyznaczonych przez formułę f_Z dla sytuacji S będziemy oznaczać $Z(S)$.

Przykład 7.2

Znajdźmy dostępne sytuacje z sytuacji przedstawionej poniżej (Tabela 20) na podstawie strategii z poprzedniego przykładu.

Sytuacja	a	b	k	l	m
S_1	2	0	2	0	1

Tabela 20 Przykładowa sytuacja

Niech $f_S = a_2 b_1 s_2 t_2 u_1 k_2 l_0 m_1$.

$$f(a_1) = (b_1 \vee b_0 \vee s_2 \vee t_1 \vee u_1 \vee m_0 \vee m_1 \vee a_2 \vee l_1) \wedge a_2 b_1 s_2 t_2 u_1 k_2 l_0 m_1 \equiv (b_1 \vee s_2 \vee u_1 \vee m_1) \wedge$$

$$a_2 b_1 s_2 t_2 u_1 k_2 l_0 m_1 \equiv a_2 b_1 s_2 t_2 u_1 k_2 l_0 m_1 \equiv f_S \text{ (formuła niesprzeczna)}$$

$$f(a_2) = (b_2 u_2 \vee a_1 u_2 l_0 \vee s_1 u_2 l_0 \vee a_1 b_2 m_2 \vee b_2 s_1 m_2 \vee a_1 l_0 m_2 \vee s_1 l_0 m_2 \vee b_2 l_2) \wedge a_2 b_1 s_2 t_2 u_1 k_2 l_0 m_1 \equiv 0$$

(sprzeczna)

$$f(b_0) = fst(b_0) \wedge f_S \equiv f_S$$

$$f(b_1) = fst(b_1) \wedge f_S \equiv f_S$$

$$f(b_2) = fst(b_2) \wedge f_S \equiv f_S$$

$$f(s_1) = fst(s_1) \wedge f_S \equiv f_S$$

$$f(s_2) = fst(s_2) \wedge f_S \equiv f_S$$

$$f(t_1) = fst(t_1) \wedge f_S \equiv f_S$$

$$f(t_2) = fst(t_2) \wedge f_S \equiv f_S$$

$$f(u_1) = fst(u_1) \wedge f_S \equiv f_S$$

$$f(u_2) = fst(u_2) \wedge f_S \equiv f_S$$

$$f(k_2) = fst(k_2) \wedge f_S \equiv f_S$$

$$f(l_0) = fst(l_0) \wedge f_S \equiv f_S$$

$$f(l_1) = fst(l_1) \wedge f_S \equiv f_S$$

$$f(l_2) = fst(l_2) \wedge f_S \equiv f_S$$

$$f(m_0) = fst(m_0) \wedge f_S \equiv 0$$

$$f(m_1) = fst(m_1) \wedge f_S \equiv f_S$$

$$f(m_2) = fst(m_2) \wedge f_S \equiv f_S$$

Strategiczna przestrzeń sytuacji w rozpatrywanej sytuacji będzie następująca:

$$f_Z(S) = a_1 \wedge (b_0 \vee b_1 \vee b_2) \wedge (s_1 \vee s_2) \wedge (t_1 \vee t_2) \wedge (u_1 \vee u_2) \wedge k_2 \wedge (l_0 \vee l_1 \vee l_2) \wedge (m_1 \vee m_2) \wedge f_\emptyset$$

Strategiczna przestrzeń sytuacji S , ze względu na ograniczenia, składa się z 66 sytuacji.

7.3 Analiza ruchów

Analizując strategiczną przestrzeń sytuacji możemy wyznaczyć *strategiczne przestrzenie ruchów* agentów czyli zbiór stanów lokalnych, osiągalnych z wybranego stanu lokalnego w danej strategii.

Niech $Z(S)$ będzie strategiczną przestrzenią dla sytuacji S a f_S formułą opisującą sytuację S :

$$f_S = \bigwedge_{ag \in Ag} f_{sag}^S$$

Strategiczna przestrzeń ruchów agenta ag jest opisana formułą

$$f_r(ag) = \bigvee_{S' \in Z(S)} f_{sag}^{S'}$$

gdzie f_{sag}^S jest opisem stanu lokalnego s agenta ag w sytuacji S .

Na przykład w bieżącej sytuacji agent ag_1 może zmienić stan a_2b_1 na stan a_1b_0 lub a_1b_1 lub a_1b_2 (patrz Przykład 7.2).

Osobnym problemem jest, który ze stanów ze strategicznej przestrzeni ruchów agent wybierze.

Najprostszym rozwiązaniem wydaje się odniesienie analizowanych stanów do zbioru celów danego agenta. Im dany stan jest bliższy stanowi ze zbioru celów tym jest bardziej preferowany i tym pewniej zostanie on wybrany w kolejnym ruchu agenta. Odniesienie do zbioru celów polega najpierw na sprawdzeniu czy któryś z rozpatrywanych stanów nie należy do zbioru celów. Sprawdzenie takie sprowadza się do utworzenia koniunkcji formuły określającej zbiór celów danego agenta z formułą opisującą przestrzeń ruchów agenta. Implikanty pierwsze z takiej koniunkcji wyznaczają wszystkie stany będące celem agenta i osiągalne w danej sytuacji. Jeżeli żaden rozpatrywany stan nie należy do zbioru celów to możemy szukać, który z tych stanów jest najbliższy do stanu ze zbioru celów: obliczyć

najmniejszą odległość danego stanu od zbioru celów (funkcja odległości 3.1.4). Jednakże wartość odległości może nie być dobrym wskaźnikiem dla stanów dalekich od zbioru celów.

Innym sposobem jest statystyczna analiza tablicy danych historycznych. Można założyć, że stany najczęściej powtarzające się w tej tablicy to stany bardziej preferowane przez agentów. Z rozpatrywanych stanów agent najprawdopodobniej wybierze taki, który częściej od pozostałych występował w tablicy danych historycznych. Dodatkowo każdy algorytm wyboru stanu powinien uwzględniać i minimalizować koszt zmiany stanu.

Przykład 7.3

Strategiczna przestrzeń ruchów agenta ag_1 jest opisana formułą:

$$f_r(ag_1) = a_1b_0 \vee a_1b_1 \vee a_1b_2$$

Weryfikując przestrzeń ruchów agenta ag_1 z jego zbiorem celów otrzymujemy:

$$t_{ag_1} \wedge f_r(ag_1) = (a_2 \vee b_2) \wedge (a_1b_0 \vee a_1b_1 \vee a_1b_2) = a_1b_2$$

Stan a_1b_2 jest stanem ze zbioru celów, dlatego możemy przyjąć, że zostanie on wybrany przez agenta ag_1 .

Badając częstość występowania stanów w danych historycznych wybralibyśmy stan a_1b_0 jako preferowany przez agenta ag_1 (7 przypadków). Ze względu na koszt zmiany stanu, najbardziej preferowanym stanem (najmniejszy koszt zmiany stanu) jest również stan a_1b_0 – wartość kosztu równa 1/2.

7.4 Przewidywanie konfliktu, zagrożenie konfliktem

Stan zagrożenia występuje wtedy, gdy strategie agentów prowadzą do konfliktu globalnego. Aby wykryć stan zagrożenia po każdym ruchu agentów możemy weryfikować strategiczną przestrzeń sytuacji z sytuacjami dobrymi. Jeżeli liczba sytuacji dobrych w strategicznej przestrzeni sytuacji jest mniejsza od zadanego poziomu to potencjalnie występuje zagrożenie konfliktem globalnym. Dokładniej wskaźnik

$$cr(S) = \frac{card(Zcore(S))}{card(Z(S))}$$

gdzie $Zcore(S)$ jest zbiorem *sytuacji dobrych* w strategicznej przestrzeni sytuacji, jest *wskaźnikiem zagrożenia konfliktem* w sytuacji S . Jeżeli natomiast wskaźnik $cr(S)$ jest większy od zadanego poziomu to mówimy, że sytuacja S jest *bezpieczna*. Jeśli natomiast wskaźnik $cr(S)$ jest mniejszy od zadanego poziomu to sytuacja jest *niebezpieczna*.

Zbiór *sytuacji dobrych* $Zcore(S)$ w strategicznej przestrzeni sytuacji wyznacza następująca formuła:

$$f_{zc}(S) = f_c \wedge f_z(S)$$

gdzie f_c oznacza formułę opisującą zbiór sytuacji dobrych a $f_z(S)$ opisuje strategiczną przestrzeń sytuacji dla sytuacji S . Implikanty pierwsze formuły $f_{zc}(S)$ definiują zbiór sytuacji dobrych $Zcore(S)$.

Przykład 7.4

Przyjmijmy wskaźnik $cr(S) < 0,3$ za stan zagrożenia konfliktem oraz $cr(S) > 0,7$ za stan bezpieczny w naszym przykładzie.

Poprzednio (Przykład 7.2) wyznaczyliśmy strategiczną przestrzeń dla bieżącej sytuacji. Natomiast zbiór sytuacji dobrych w tym konflikcie jest zdefiniowany z pomocą formuły:

$$f_c = a_1k_2l_2m_2 \vee a_1s_2k_2l_2m_1 \vee b_2s_2k_2l_2m_1 \vee a_1u_2k_2l_2m_1 \vee b_2u_2k_2l_2m_1 \vee a_1t_1k_2l_2m_1 \vee b_2t_1k_2l_2m_1 \vee k_1l_1m_1 \\ \vee k_1l_2m_2 \vee t_1u_2k_2l_2m_1 \vee s_1u_2k_2l_2m_1$$

Zbiór sytuacji dobrych w strategicznej przestrzeni sytuacji:

$$f_{zc}(S) = a_1b_0s_2t_1u_1k_2l_2m_1 \vee a_1b_0s_2t_1u_2k_2l_2m_1 \vee a_1b_0s_1t_2u_2k_2l_2m_1 \vee a_1b_1s_2t_1u_1k_2l_2m_1 \vee a_1b_1s_2t_1u_2k_2l_2m_1 \\ \vee a_1b_1s_1t_2u_2k_2l_2m_1 \vee a_1b_2s_2t_1u_1k_2l_2m_1 \vee a_1b_2s_2t_1u_2k_2l_2m_1 \vee a_1b_2s_1t_2u_2k_2l_2m_1 \vee a_1b_0s_2t_1u_1k_2l_2m_2 \vee \\ a_1b_0s_2t_1u_2k_2l_2m_2 \vee a_1b_0s_1t_2u_1k_2l_2m_2 \vee a_1b_0s_1t_2u_2k_2l_2m_2 \vee a_1b_1s_2t_1u_1k_2l_2m_2 \vee a_1b_1s_2t_1u_2k_2l_2m_2 \vee \\ a_1b_1s_1t_2u_1k_2l_2m_2 \vee a_1b_1s_1t_2u_2k_2l_2m_2 \vee a_1b_2s_2t_1u_1k_2l_2m_2 \vee a_1b_2s_2t_1u_2k_2l_2m_2 \vee a_1b_2s_1t_2u_1k_2l_2m_2 \vee \\ a_1b_2s_1t_2u_2k_2l_2m_2$$

$$cr(S)=21/66$$

Ponieważ $0,3 < cr(S) < 0,7$ sytuacja S nie jest sytuacją bezpieczną ani sytuacją grożącą konfliktem.

7.5 Rekonstrukcja sytuacji grożących konfliktem

Analizując strategiczne przestrzenie sytuacji możemy utworzyć *graf zmienności sytuacji konfliktowej*. Węzłami grafu są sytuacje. Każda sytuacja jest otoczona swoją strategiczną przestrzenią sytuacji. Gałęzie grafu łączą wszystkie dostępne sytuacje i są skierowane od sytuacji wyjściowej do osiągalnej zgodnie ze strategią. Wielkość (głębokość) grafu maksymalnie odpowiada liczbie sytuacji osiągalnych (graf zdegenerowany). Analizując graf zmienności sytuacji konfliktowych możemy nie tylko ostrzegać przed sytuacjami z zagrożeniem konfliktem czy cyklami (powtarzaniem sytuacji) ale również możemy wyznaczyć najkrótszą czy minimalną ze względu na koszt zmiany stanu drogę (sposób) rekonstrukcji sytuacji konfliktowej do sytuacji bezpiecznej.

Podstawowy sposób postępowania przy rekonstrukcji bieżącej sytuacji do zadanej zgodnie z daną strategią polega na budowie grafu a potem jego przeszukiwaniu. Poczynając od bieżącej sytuacji wyznaczamy strategiczną przestrzeń sytuacji dla węzłów grafu. Tworząc nowy węzeł sprawdzamy czy osiągnęliśmy już zadaną sytuację. Przeglądanie grafu może być oparte na algorytmach przeglądania wzdłużnego (ang. breadth-first). Szczegółowe analizy w grafie mogą być uciążliwe ze względu na zazwyczaj dużą liczbę sytuacji osiągalnych występujących w rzeczywistych konfliktach.

STRATEGIE OBLICZENIOWE

W pracy zastosowano wnioskowanie boolowskie do opisu problemów i ich rozwiązywania. Metoda ta pozwala w przejrzysty sposób opisać skomplikowane zadania sztucznej inteligencji i je rozwiązywać w oparciu o proste metody logiczne. Niestety w rzeczywistych konfliktach formuły boolowskie mogą być bardzo duże a operacje na nich stwarzają problemy obliczeniowe (czas i pamięć). W tym rozdziale przyjrzymy się bliżej zastosowanym algorytmom i zaproponujemy proste i bardziej wyszukane strategie obliczeniowe.

8.1 Wyznaczanie minimalnych reguł decyzyjnych

Punktem wyjścia do rozwiązywania wielu problemów przedstawionych w pracy są tablice decyzyjne (zbiory celów 3.1.5, sytuacje dobre 3.4.1.1). Na ich podstawie poprzez tablice rozróżnialności wyznaczaliśmy minimalne reguły decyzyjne (algorytm przedstawiono w rozdziale 12.4 a przykład zaprezentowano w punkcie 3.1.6). Sposób wyznaczania reguł decyzyjnych jest szeroko opisany w literaturze [29], [44]. Wyznaczenie minimalnych reduktów tablicy decyzyjnej jest problemem NP – trudnym. Opracowaniem heurystyk dla tego problemu zajmowało się wielu autorów. W jednym z nich zastosowano algorytmy genetyczne [69].

8.2 Obliczanie implikantów pierwszych

Rozwiązanie problemu opisanego formułą boolowską polega na redukcji formuły do postaci dyzjunkcyjnej poprzez wykonywanie koniunkcji i zastosowanie prawa pochłaniania dopóki to możliwe. Ogólnie problem ten jest o wykładniczej złożoności obliczeniowej [44]. Warto więc zastanowić się nad zastosowaniem heurystyk jeśli rozmiar formuł jest tak duży, że natrafimy na problemy ograniczeń pamięci lub ograniczeń czasowych. Poniżej autor prezentuje cztery proste takie strategie (heurystyki), które skracają proces obliczeń. Zaproponowane heurystyki sprowadzają się do redukcji wielkości formuły boolowskiej.

Niestety, zaproponowanie algorytmu aproksymacyjnego o złożoności wielomianowej pozostaje problem na razie nierozwiązanym.

W większości prezentowanych w pracy problemów musimy zredukować formułę opisującą ograniczenia (duża formuła f_ϕ) z formułami opisującymi stany dobre f_c i "małymi" formułami pochodzącymi od agentów biorących udział w konflikcie (w różny sposób wyznaczone cele agentów czy ich preferencje). Ogólnie formuła ma zazwyczaj następującą postać:

$$f = f_1 \wedge \dots \wedge f_n \wedge f_c \wedge f_\phi$$

gdzie $f_1 \dots f_n$ są formułami opisującymi wymagania agentów.

Ważnym spostrzeżeniem, które można wykorzystać podczas redukcji formuły f jest fakt, że wynik (jeżeli istnieje) jest dyzjunkcją wybranych komponentów formuły f_ϕ . Natomiast formuły $f_1 \dots f_n$ z założenia zawierają komponenty oparte o różne zmienne boolowskie – zbiory atrybutów $\{A_{ag}\}$ są parami rozłączne dlatego ich koniunkcja nie powoduje redukcji.

8.2.1 Proste metody redukcji formuł boolowskich

Redukując formułę boolowską można zastosować prawa algebry Boola dla zwiększenia szybkości obliczeń (zmniejszenia liczby kroków niezbędnych do redukcji). Przede wszystkim należy zwrócić uwagę na dobór podformuł, na których wykonana będzie koniunkcja. Ze względu na prawo pochłaniania należy tak dobrać formuły aby możliwie jak najwięcej komponentów uległo redukcji. W naszym przypadku należy najpierw mnożyć formuły f_1, \dots, f_n z formułą f_ϕ (algorytm doboru formuł przedstawiono w punkcie 8.2.2).

W przypadku, gdy zbiory wartości atrybutów są małe może być warto zamienić komponent na odpowiadającą dyzjunkcję negacji stosując prawa de Morgana. Rozważmy dla przykładu formułę $f = (a_2 b_2 \vee a_2 b_1) \wedge (a_2 b_0 \vee a_1 b_0)$. Zakładając, że $\overline{b_0} = b_2 \vee b_1$ oraz $\overline{a_0} = a_2 \vee a_1$ dyzjunkcję $a_2 b_2 \vee a_2 b_1 \equiv a_2 (b_2 \vee b_1)$ można zamienić na $a_2 \overline{b_0}$, a formułę

$(a_2 b_0 \vee a_1 b_0)$ na $b_0 \overline{a_0}$. Otrzymujemy więc formułę $f = b_0 \overline{a_0} \wedge a_2 \overline{b_0} \equiv \emptyset$, która jest sprzeczna. Aby przekonać się o tym nie używając negacji należałoby wyznaczyć cztery koniunkcje.

8.2.2 Strategia mnożenia formuł

Jak wcześniej wspomniano strategia ta opiera się na spostrzeżeniu, że małe formuły mogą szybko zredukować duże. Warunkiem koniecznym (choć niewystarczającym) jest aby formuły używały tych samych zmiennych boolowskich. Można również zauważyć, że mnożenie formuł opisujących ograniczenia w celu uzyskanie jednej formuły f_ϕ opisującej sytuacje osiągalne jest nieefektywne. Formuły opisujące ograniczenia muszą ulec redukcji poprzez nałożenie na nie preferencji agentów. Strategię doboru podformuł do mnożenia tak, by zwiększyć szybkość redukcji formuły, można opisać w następujący sposób:

Niech $f[]$ – reprezentuje tablicę formuł

for $i=1$ **to** liczba formuł **do**

$vnComponents[i]$ = liczba komponentów formuły $f[i]$

$vsAttr[i]$ = łańcuch znaków - atrybuty występujące w formule $f[i]$

$vnAttr[i]$ = liczba różnych atrybutów użytych w formule $f[i]$

zaznacz formułę $f[i]$ jako nieużywaną

endfor;

for $i=1$ **to** liczba formuł **do**

if $f[i]$ używana **then**

continue;

for $j=i+1$ **to** liczba formuł **do**

if $f[j]$ używana **then**

continue;

if $vsAttr[i] \subseteq vsAttr[j]$ **and** $max(vnComponents[j])$ **then**

$f_i = f_j \wedge f_i$ // oblicz koniunkcję formuł

wyznacz nową wartość $vnComponents[j]$, $vsAttr[j]$ i $vnAttr[j]$,

zaznacz formułę f_j jako użytą,

break;

endif;

```

    endfor;
endfor;
z = 0;
for i=1 to liczba formuł do
    if not f[i] używana then
        if z == 0 then
            z = i;
            continue;
        endif;
        f[z]=f[z] ∧ f[i]    // oblicz koniunkcję formuł
    endif;
endfor;
drukuj f[z]; // zredukowana formuła wejściowa

```

Złożoność obliczeniowa przedstawionego algorytmu nie jest mniejsza niż algorytmu polegającego na kolejnym mnożeniu formuł i gdy mnożenia nie przynoszą redukcji złożoność jest wykładnicza. Jeżeli jednak komponenty ulegają redukcji to w zaproponowanym algorytmie odbywa się to wcześniej niż w tradycyjnej metodzie.

8.2.3 Strategia wykluczania (heurystyka)

Zaproponowane w tym i w następnym punkcie strategie redukcji formuł boolowskich przyspieszają proces poszukiwania pierwszego, czy pierwszych rozwiązań (nie prowadzą w łatwy sposób do znalezienia wszystkich rozwiązań). Ogólnie strategia wykluczania opiera się na wstępnej redukcji komponentów formuł na podstawie komponentów agentów. Formuły pochodzące od agentów są postaci dyzjunkcji komponentów. Można zauważyć, że jeśli wynik istnieje to musi zawierać, któryś z komponentów każdego agenta (co najmniej jeden). Dlatego proponujemy w pierwszym kroku zredukować wszystkie komponenty na podstawie nie wybranych komponentów danego agenta. Strategia ta jest heurystyką, ponieważ rozwiązanie może zawierać więcej niż jeden komponent pochodzący od danego agenta. Redukując komponenty w ten sposób możemy usunąć również niektóre (bądź wszystkie) rozwiązania. Strategię tę można opisać w następujący sposób:

Niech $f[i][j]$ będzie dwuwymiarową tablicą przechowującą formuły – dla każdej formuły są pamiętane jej komponenty; niech pierwsze nAg formuł tablicy f pochodzi od agentów $iC[i]$ – jest tablicą indeksów wskazujących wybrany komponent dla każdego agenta, który nie ulega wykluczeniu

$z = 0$;

for $i=1$ **to** liczba nAg **do**

for $j=1$ **to** liczba komponentów $f[i]$ **do**

if $j == iC[i]$ **then** // pomiń komponent, który nie zostanie wykluczony

continue;

$z++$;

$vsetAgComp[z] = f[i][j]$; //zapamiętaj j -ty komponent formuły i -tego agenta

endfor;

endfor;

for $i=1$ **to** liczba formuł **do**

for $j=1$ **to** liczba komponentów $f[i]$ **do**

for $z=1$ **to** rozmiar $vsetAgComp$ **do**

if $vsetAgComp[z] \subseteq f[i][j]$ **then**

 usuń komponent $f[i][j]$

endfor;

endfor;

endfor;

// wykonaj koniunkcję pozostałych po redukcji formuł

for $i=1$ **to** liczba formuł-1 **do**

$f[\text{liczba formuł}] = f[i] \wedge f[\text{liczba formuł}]$;

endfor;

drukuj $f[\text{liczba formuł}]$; // wynik - zredukowana formuła

Jeśli formuła jest sprzeczna (nie znaleziono rozwiązania) można wybrać inne komponenty – inaczej wypełnić tablicę $iC[i]$ i powtórzyć algorytm.

Sprawdzenie czy dany komponent formuły zawiera inny wybrany komponent polega na wyznaczeniu ich koniunkcji. Natomiast w przeciwieństwie do tradycyjnego algorytmu mnożenia formuł nie zapamiętujemy wyniku koniunkcji lecz jeśli dany komponent zawiera inny to go usuwamy, jeśli zaś nie zawiera to pozostawiamy komponent bez zmian. Złożoność obliczeniowa zaproponowanego algorytmu redukuje się do złożoności liniowej (zależnej od liczby agentów) w tej części, która jest koniunkcją formuł pochodzących od agentów. Dalsze obliczenia (koniunkcja) prowadzi do algorytmu o pesymistycznym czasie działania $O(n^p)$ dla n formuł i p komponentów w każdej formule.

Aby zwiększyć szansę znalezienia rozwiązania danego problemu można powtórzyć algorytm dla różnych możliwych kombinacji i różnej liczby wybranych komponentów od każdego agenta. Osobnym problemem jest taki dobór komponentów aby jak najszybciej otrzymać rozwiązanie. Ponieważ istnieje obawa, że w wyniku algorytmu zbyt wiele komponentów ulegnie wykluczeniu, należy tak dobierać komponenty agentów aby pozostałe powodowały jak najmniejsze redukcje. Proponuje się na przykład zostawić krótsze komponenty bo dłuższe słabiej redukują pozostałe formuły. Innym kryterium doboru może być częstość występowania atrybutów w pozostałych komponentach (im częściej występują tym prawdopodobnie mają większą szansę znalezienia się w rozwiązaniu problemu).

Przykład 8.1

Zastosujmy opisaną strategię do problemu poszukiwania konsensusu na poziomie globalnym i lokalnym w konflikcie między pracodawcą i pracownikami. Formuła opisująca ten problem ma postać (patrz punkt 5.2):

$$f = (a_2 \vee b_2) \wedge (s_2 t_1 \vee s_2 t_2 \vee s_2 u_2) \wedge (l_2 \vee k_2 m_2) \wedge f_C \wedge f_\varphi$$

gdzie formuły f_C i f_φ oznaczają odpowiednio formułę opisującą sytuację dobre i ograniczenia. Formuły f_C i f_φ nie rozwinęliśmy tutaj ze względu na ich wielkość (formuła f_φ ma około 170 komponentów).

Jeśli dla pierwszego agenta wybierzemy komponent a_2 to po redukcji i wykonaniu mnożenia formuła nie ma rozwiązań – jest to zgodne z otrzymanymi wynikami tradycyjną

metodą (rozwiązania nie zawierają deskryptora a_2). Wybierzmy więc komponent b_2 (natomiast komponent a_2 ulegnie redukcji). Dla drugiego agenta wybierzmy pierwszy komponent s_2t_1 i dla trzeciego l_2 . Po przeprowadzeniu redukcji, zgodnie z algorytmem, usunęliśmy około 1/3 komponentów, ponadto formuła ma teraz znacznie łatwiejszą postać bo niektóre dyzjuncje uległy redukcji do postaci komponentów. Po wykonaniu koniunkcji otrzymujemy ostatecznie 2 implikanty pierwsze:

$$f = a_1b_2s_2t_1u_1k_2l_2m_1 \vee a_1b_2s_2t_1u_2k_2l_2m_1$$

Otrzymane rozwiązanie jest podzbiorem rozwiązania otrzymanego tradycyjną metodą. Aby otrzymać pełne rozwiązanie należałoby nie redukować żadnego komponentu agenta ag_3 .

8.2.4 Strategia wykluczania 2 (heurystyka)

Przedstawiona poniżej strategia jest modyfikacją strategii opisanej w poprzednim punkcie. Ma ona szczególne znaczenie przy odpowiedniej, komputerowej reprezentacji danych w problemie obliczania implikantów pierwszych. Szczegóły implementacyjne zastosowane przez autora są opisane w rozdziale poświęconym programowi analizy konfliktów (rozdział 9.2.3). Ogólnie strategia opiera się na wstępnej redukcji komponentów formuł na podstawie użytych atrybutów bez uwzględniania wartości atrybutów. Podobnie jak poprzednio punktem wyjścia są formuły pochodzące od agentów. Strategię można opisać w poniższy sposób. Przyjmujemy następujące oznaczenia.

$f[][]$ – dwuwymiarowa tablica przechowująca formuły – dla każdej formuły są pamiętane jej komponenty; niech pierwsze nAg formuł tablicy f pochodzi od agentów

$iC[]$ – tablica indeksów wskazujących wybrany komponent dla każdego agenta, który nie zostanie wykluczony

$vsetAttr[], vsetAgAttr[]$ – tablice reprezentujące zbiory atrybutów

Set $GetAttr(Component)$ – funkcja wyznaczająca zbiór atrybutów z komponentu

$z = 0;$

for $i=1$ **to** nAg **do**

```

for j=1 to liczba komponentów  $f[i]$  do
    if j ==  $iC[i]$  then // zapamiętaj atrybuty wybranego komponentu
         $vsetAgAttr[z] = GetAttr(f[i][j] )$ ;
        continue;
    endif;
    z++;
endfor;
endfor;
z = 0;
for i=1 to  $nAg$  do
    for j=1 to liczba komponentów  $f[i]$  do
        if j ==  $iC[i]$  then // pomiń komponent, który (został wybrany)
            continue;
        if (  $GetAttr(f[i][j]) \cap vsetAgAttr[ iC[i] ] = \emptyset$  ) then
            z++;
             $vsetAttr[z] = GetAttr(f[i][j] )$ ;
        endif;
    endfor;
endfor;
for i=1 to liczba formuł do
    for j=1 to liczba komponentów  $f[i]$  do
        if  $j \leq nAg$  and j ==  $iC[i]$  then
            continue; // pomiń komponent, który nie został wykluczony
        for k=1 to rozmiar  $vsetAttr$  do
            if (  $GetAttr(f[i][j]) \cap vsetAttr[k] \neq \emptyset$  ) then
                for z=1 to liczba agentów do
                    if (  $f[i][j] \cap vsetAgAttr[z] = \emptyset$  ) then
                        break;
                    endif;
                endfor;
            if z <= liczba agentów then
                usuń komponent  $f[i][j]$ ;
            endif;
        endfor;
    endfor;

```

```

                break; // weź następny komponent
            endif;
        endif;
    endfor;
endfor;
// wykonaj koniunkcję pozostałych po redukcji formuł np.
for i=1 to liczba formuł-1 do
     $f[\text{liczba formuł}] = f[i] \wedge f[\text{liczba formuł}]$ 
endfor;
drukuj  $f[\text{liczba formuł}]$ ; // wynik – pierwsze implikanty formuły wejściowej  $f$ 

```

Oczywiście i ten algorytm można powtórzyć dla różnych możliwych kombinacji i liczby wybranych komponentów dla wszystkich agentów. Sposób wyboru komponentów stosujemy taki sam jak w poprzednim punkcie. Należy zwrócić uwagę na niebezpieczeństwo usunięcia całej formuły opisującej ograniczenia, czy pozbycie się w ogóle jakiegoś atrybutu. Otrzymany wtedy wynik jest tylko częściowym rozwiązaniem problemu. Jak zostało wspomniane pewne szczegóły implementacyjny zachęcają do zastosowania powyższego algorytmu. Mianowicie każdy komponent formuły w programie autora pracy reprezentowany jest przez bitową tablicę atrybutów użytych w komponencie i tablicę wartości tych atrybutów. Koniunkcja komponentów polega na logicznej sumie tablic bitowych (jeden rozkaz). Dodatkowo należy zweryfikować wartości atrybutów użytych w obu komponentach. Logiczny iloczyn tablic bitowych mówi nam o tym które atrybuty się powtarzają. Należy dla tych atrybutów sprawdzić wartości – jeśli choć dla jednego atrybutu wartości są różne to cały komponent można odrzucić. Stosując powyższy algorytm możemy wstępnie odrzucić część komponentów bez wykonywania dodatkowego sprawdzania wartości atrybutów. (A to właśnie ten dodatkowy krok jest bardziej czasochłonny.)

Przykład 8.2

Przeanalizujemy działanie algorytmu na tym samym problemie szukania konsensusu na poziomie globalnym i lokalnym. Formuła opisująca problem została przedstawiona powyżej

(Przykład 8.1). Tak jak poprzednio wybierzemy następujące komponenty i atrybuty dla każdego agenta: ag_1 : b dla ag_2 : s, t i dla ag_3 : l . Oznacza to, że redukcji powinien ulec komponent zawierający atrybut a i nie zawierający atrybuty: b lub s, t lub l itd. Tym razem redukcji uległo około $\frac{1}{4}$ komponentów. Ostatecznie otrzymamy identyczny wynik jak poprzednio:

$$f = a_1 b_2 s_2 t_1 u_1 k_2 l_2 m_1 \vee a_1 b_2 s_2 t_1 u_2 k_2 l_2 m_1$$

8.2.5 Strategia wstępnej redukcji

Podobnie jak w strategiach wykluczania strategia ta opiera się na wstępnej redukcji komponentów formuł na podstawie komponentów formuł agentów. Formuły pochodzące od agentów są postaci dyzjunkcji komponentów. W problemach konsensusu jeśli wynik (sytuacja niekonfliktowa) istnieje to musi zawierać któryś z komponentów każdego agenta (co najmniej jeden). W strategii wstępnej redukcji przeglądamy każdy komponent pozostałych formuł i weryfikujemy go z komponentami agentów. Jeżeli rozważany komponent nie zawiera żadnego komponentu formuły opisującej zbiór celów danego agenta to możemy go usunąć (nie wchodzi do rozwiązania). Dokładniej strategię można opisać z pomocą następującego algorytmu:

Niech $f[][]$ będzie dwuwymiarową tablicą przechowującą formuły – dla każdej formuły są pamiętane jej komponenty i niech pierwsze nAg formuł tablicy $f[][]$ pochodzi od agentów.

```

for  $i=nAg$  to liczba formuł do
    for  $j=1$  to liczba komponentów  $f[i]$  do
         $bRemove = false$ ;
        for  $z=1$  to  $nAg$  do
            for  $k=1$  to liczba komponentów  $f[z]$  do // dla jednego agenta
                if  $f[i][j] \wedge f[z][k] = f[z][k]$  then
                    break; // nie usuwaj komp.
                if  $f[i][j] \wedge f[z][k] = \emptyset$  then
                     $bRemove = true$ ;
        endfor;

```

```

        if k <= liczba komponentów f[z] then
            break;
        endfor;
    if bRemove then
        usuń komponent f[i][j];
    endif;
endfor;
endfor;
// wykonaj koniunkcję pozostałych po redukcji formuł
for i=1 to liczba formuł-1 do
    f [liczba formuł] = f[i] ∧ f[liczba formuł];
endfor;
drukuj f [liczba formuł]; // wynik - zredukowana formuła

```

Strategia wstępnej redukcji formuł pozwala w czasie zależnym od liczby komponentów formuł agentów i liczby komponentów pozostałych formuł (pesymistyczny czas działania $O(n^2)$) sprawdzić i ewentualnie zredukować te komponenty, które i tak uległyby redukcji podczas wykonywania koniunkcji. W algorytmie bez wstępnej redukcji może następować wykładniczy wzrost liczby komponentów zanim zostaną one zredukowane. Za pomocą zaproponowanego algorytmu nie usuwamy jednak wszystkich komponentów, które ulegną redukcji.

Przykład 8.3

Rozważmy dla przykładu problem konsensusu na poziomie globalnym i lokalnym. Formuła opisująca problem jest następująca:

$$f = (a_2 \vee b_2) \wedge (s_2 t_1 \vee s_2 t_2 \vee s_2 u_2) \wedge (l_2 \vee k_2 m_2) \wedge (a_1 k_2 l_2 m_2 \vee a_1 s_2 k_2 l_2 m_1 \vee b_2 s_2 k_2 l_2 m_1 \vee a_1 u_2 k_2 l_2 m_1 \vee b_2 u_2 k_2 l_2 m_1 \vee a_1 t_1 k_2 l_2 m_1 \vee b_2 t_1 k_2 l_2 m_1 \vee k_1 l_1 m_1 \vee k_1 l_2 m_2 \vee t_1 u_2 k_2 l_2 m_1 \vee s_1 u_2 k_2 l_2 m_1) \wedge f_\phi$$

Pokażemy działanie algorytmu na przykładzie formuły opisującej sytuację dobre. Ponieważ pierwszy komponent: $a_1 k_2 l_2 m_2$ zawiera komponent l_2 agenta ag_3 nie może on zostać usunięty. Podobnie z następnymi komponentami. Jedynie komponent $k_1 l_1 m_1$ zostanie

usunięty ponieważ $k_1 l_1 m_1 \wedge (l_2 \vee k_2 m_2) = \emptyset$. Oczywiście można sobie wyobrazić formułę, w której zysk z zastosowania algorytmu jest większy (więcej komponentów ulegnie redukcji).

8.2.6 Scalanie agentów (*clustering*)

Znajomość koalicji może pomóc w rozwiązywaniu problemu konsensusu. W konfliktach, w których bierze udział duża liczba agentów może być warto podzielić konflikt na mniejsze podkonflikty i rozwiązywać je osobno. Jednym ze sposobów jest rozwiązanie najpierw "słabych" konfliktów w koalicjach. Znajomość koalicji możemy wykorzystać też w strategii obliczeniowej polegającej na "scalaniu" agentów (ang. clustering [17]). Scalanie polega na reprezentacji koalicji wielu agentów z pomocą jednego (nowego) agenta. W strategii scalania zakładamy, że rozwiązanie takiego uproszczonego konfliktu da nam aproksymacyjne rozwiązanie wyjściowego konfliktu (którego wyniki jesteśmy w stanie zaakceptować).

8.3 Programowania z ograniczeniami

Omawiane w pracy problemy, zwłaszcza problem konsensusu może być potraktowany jako problem programowania liniowego czy szerzej programowania z ograniczeniami [7], [6], [50]. Punktem wyjścia problemu są wtedy ograniczenia zadane jako nierówności, które nie przekształcamy do postaci formuł boolowskich. Dziedzina zmiennych używanych w ograniczeniach jest przeglądana i generowane są kolejne rozwiązania (sytuacje spełniające ograniczenia). Zauważmy, że inne formuły (pochodzące od agentów, czy opisujące sytuacje dobre) powinny zostać również zamienione do postaci ograniczeń (nierówności) lub wynikowe sytuacje powinny być na bieżąco weryfikowane przez eksperta lub/i agentów.

8.4 Dyskretyzacja atrybutów ciągłych

Wiele spraw rozważanych przez agentów może pochodzić z różnego typu pomiarów (np. zysk zakładu) mamy wtedy do czynienia z atrybutami ciągłymi tzn. takimi których wartości są liczbami rzeczywistymi, pochodzącymi z pewnego przedziału. System informacyjny posiadający atrybuty ciągle charakteryzuje się dużą liczbą klas równoważności, co pociąga za sobą generowanie wielu reguł decyzyjnych. Duża liczba reguł decyzyjnych daje szansę na to, że reguły są deterministyczne (choć trzeba wziąć pod uwagę błędy pomiarów).

Reguły takie jednak słabo klasyfikują nowe przypadki. Klasyfikacja nowych przypadków jest podstawą wnioskowania aproksymacyjnego, niezbędnego w analizie konfliktów. W związku z tym konieczne jest przeprowadzenie procesu dyskretyzacji atrybutów ciągłych. Dyskretyzacja wartości atrybutu ciągłego polega na podzieleniu zakresu wartości tego atrybutu na pewną liczbę przedziałów i przypisaniu tym podziałom określonych różnych wartości. Dokładniej, niech $V_a=[v_b, v_e]$ jest zakresem wartości atrybutu a oraz p_a będzie podziałem przedziału $[v_b, v_e]$ na k przedziałów: $p_a=\{[v_0, v_1],[v_1, v_2),\dots,[v_{k-1}, v_k]\}$, gdzie $v_0=v_b$, $v_k=v_e$ oraz $v_{i-1}<v_i$ dla $i=1, 2,\dots,k$. Po przeprowadzeniu procesu dyskretyzacji atrybutu ciągłego, wartości tego atrybutu są odwzorowane w dyskretne wartości atrybutu a^d , którego zbiór wartości definiuje się następująco: $V_{a^d}=\{1, 2,\dots,k\}$. Wiele metod i algorytmów dyskretyzacji atrybutów ciągłych można znaleźć w literaturze.

8.5 Operacje na tablicach decyzyjnych

Podstawową strukturą, w której są przechowywane wszystkie informacje w zaproponowanym modelu są tablice informacyjne. W wielu omawianych problemach i analizach wprowadza się atrybut decyzyjny przekształcając tablice informacyjne do tablic decyzyjnych. Na podstawie tablic decyzyjnych wyznacza się reguły decyzyjne. Ten sposób postępowania można prześledzić na przykładzie wyznaczania zbiorów celów – punkt 3.1.6. Często jednak, w rzeczywistych konfliktach mamy do czynienia z ogromną ilością informacji – dużą liczbą atrybutów czy obiektów (np. wiele stanów lokalnych czy przykładów sytuacji "dobrych" i globalnie "złych"). Dodatkowo możemy mieć do czynienia z niepewną i niepełną informacją tak, że otrzymujemy konfliktowe reguły decyzyjne lub mamy do czynienia z częściowym pokryciem. Dochodzi wtedy do sytuacji, w której dla nowo klasyfikowanego obiektu różne reguły pasujące do tego przypadku dają nam sprzeczne odpowiedzi lub przeciwnie, nie można znaleźć reguły, która mogłaby zostać zastosowana w tym przypadku. Teoria zbiorów przybliżonych daje nam metody radzenia sobie we wszystkich tych sprawach.

Podstawą redukcji liczby obiektów (przykładów) zawartych w tablicy informacyjnej jest identyfikacja klas równoważności. Ponieważ tylko jeden element jest potrzebny do reprezentacji całej klasy, każdą klasę równoważności można zastąpić jednym z jej elementów usuwając pozostałe. Przykład wyznaczania klas równoważności można znaleźć w punkcie

12.1. Jeżeli taka redukcja obiektów jest niewystarczająca możemy oprzeć się na relacji tolerancji i dokonać scalania podobnych obiektów (ang. clustering). Przykład algorytmu redukującego liczbę obiektów na podstawie relacji tolerancji znajduje się np. w [13]. Podstawą redukcji liczby atrybutów w teorii zbiorów przybliżonych jest relacja nierozróżnialności. Na jej podstawie z tablicy decyzyjnej wyznaczamy redukty (patrz punkt 12.2). Redukty są takimi minimalnymi podzbiorami atrybutów, które zachowują relację nierozróżnialności. Pozostałe atrybuty są nadmiarowe, ponieważ usunięcie ich nie powoduje pogorszenia klasyfikacji. W pracy stosuje się redukty minimalne ze względu na liczbę atrybutów. Sposób ich wyznaczania został opisany w punkcie 12.4. Bardzo dobre wyniki klasyfikacji obiektów uzyskano również dla ogromnej liczby danych stosując α -redukty 12.3. Niespójność tablicy decyzyjnej usuwa się wprowadzając uogólniony atrybut decyzyjny. Wprowadzenie atrybutu uogólnionego nie znosi sprzeczności w regułach decyzyjnych daje jednak możliwość łatwiejszego operowania na tablicy – w tym generację minimalnych, ze względu na liczbę atrybutów występujących z lewej strony, reguł decyzyjnych. Definicję i algorytm przekształcania tablicy decyzyjnej do tablicy decyzyjnej spójnej można znaleźć np. w [29]. Teoria zbiorów przybliżonych, choć nie rozstrzyga problemów klasyfikacji to rozszerza pojęcie przynależności do zbiorów w ten sposób, że możemy zaklasyfikować obiekt jako taki, o którym z pewnością nie możemy powiedzieć ani, że należy ani, że nie należy do danej klasy równoważności. Zbiór takich obiektów nazywamy brzegiem zbioru klasyfikowanych obiektów, a jego interpretacja zależy od konkretnych zastosowań. Natomiast na poziomie reguł w systemach decyzyjnych niezbędne jest rozstrzygnięcie konfliktów. Powstało wiele metod, które automatyzują wybór reguły decyzyjnej w sytuacji konfliktów. Należą do nich metody oparte na jakości reguł (ang. quality). Poszczególnym regułom przyporządkowuje się liczby zgodnie z przyjętą formułą jakości. Następnie dokonuje się wyboru tej reguły, która rozpoznaje obiekt i ma najwyższą jakość. Inną znaną metodą jest tak zwane głosowanie (ang. voting). W metodzie tej z każdym zbiorem reguł wskazujących na określoną klasę decyzyjną wiążemy pewną wagę (liczbę głosów). Wartość decyzji, na którą padło najwięcej głosów jest przyjmowana jako ostateczna. W innej pracy [63] zaproponowano wykorzystanie prostych struktur sieci neuronowych do rozstrzygnięcia konfliktów. Bardziej szczegółowe omówienie problemów rozwiązywania konfliktów pojawiających się między regułami decyzyjnymi można znaleźć np. w [63].

PROGRAM WSPOMAGANIA ANALIZY KONFLIKTÓW

Ważnym wynikiem pracy jest również opracowany komputerowy program wspomagania analizy konfliktów. Program służy do analizy konfliktów w oparciu o zaproponowany w pracy model. Jest on przeznaczony do pracy w środowisku Windows, choć modułarna budowa programu, wydzielenie interfejsu użytkownika i zastosowanie standardowych bibliotek (STL) pozwala na łatwe przeniesienie programu do systemu operacyjnego UNIX. Wszystkie przykłady zostały obliczone z pomocą programu.

9.1 Moduły programu

Program został zaprojektowany w ten sposób, aby można było wykorzystać zaimplementowane w nim algorytmy do rozwiązywania innych problemów (do rozwiązania, których można zastosować wnioskowanie boolowskie). Oparty jest on na trzech głównych modułach pokrywających funkcjonalność programu: module wyznaczania minimalnych reguł decyzyjnych na podstawie tablicy decyzyjnej, module przekształcania zależności w formuły boolowskie (dyskretne programowanie liniowe) oraz na module redukcji formuł boolowskich (do normalnej postaci dyzjunkcyjnej).

Moduł wyznaczania minimalnych reguł decyzyjnych działa zgodnie z algorytmem przedstawionym w rozdziale 12.4. Danymi wejściowymi modułu jest tablica decyzyjna i wartość atrybutu decyzyjnego. Dla zadanej wartości atrybutu decyzyjnego moduł znajduje opis klasy decyzyjnej za pomocą reguł minimalnych i zapisuje wynik w pliku tekstowym o odpowiednim formacie. (Sposób wyznaczenia minimalnych ze względu na liczbę użytych atrybutów reguł decyzyjnych jest przedstawiony w punkcie 12.4.) Tablica decyzyjna również jest przechowywana w pliku tekstowym. W nagłówku tego pliku określa się rozmiar tablicy decyzyjnej (liczbę wierszy i liczbę kolumn). Kolejne wartości w wierszach są oddzielone spacjami lub tabulatorami. Z założenia ostatnia kolumna odpowiada wartościom atrybutu

decyzyjnego. Moduł ten został opracowany na podstawie powstałej na Uniwersytecie Warszawskim biblioteki algorytmów RSES-lib.

Wprowadzenie do modelu sytuacji konfliktowej ograniczeń spowodowało do użycia metod z programowania matematycznego w tej części definiowania konfliktu. Zapis ograniczeń w postaci równań, nierówności wydaje się bardziej intuicyjny niż w postaci formuły boolowskiej. Powstała więc potrzeba stworzenia modułu programowego przekształcającego wyżej wymienione zależności do postaci formuły boolowskiej. (Prosty przykład takiego przekształcenia przedstawiono w punkcie 3.3.1.) Stworzony przez autora moduł skanuje dziedziny atrybutów (zmiennych) zapamiętując wartości spełniające ograniczenia. Powstały komponent formuły boolowskiej (zmiennymi boolowskimi są pary atrybut, wartość) dopisuje się do pliku wynikowego. Ponieważ zastosowano przegląd zupełny to złożoność algorytmu można oszacować na $O(n^p)$ (jeśli n oznacza liczbę użytych zmiennych w zależności a p licznosc zbioru wartości każdej zmiennej). W zależnościach dopuszcza się używanie podstawowych operatorów matematycznych: mnożenia (*), dzielenie (/), dodawania, odejmowania, warunków: równy, mniejszy, większy, mniejszy równy (\leq), większy równy (\geq), różny (\neq) oraz zastosowanie dowolnych stałych liczbowych.

Trzeci moduł programu jest odpowiedzialny za redukcję formuły boolowskiej. Wszystkie problemy omawiane w pracy są kodowane w ten sposób aby obliczenie implikantów pierwszych formuły boolowskiej było ich rozwiązaniem. Zadaniem modułu jest więc wyznaczenie wszystkich implikantów pierwszych formuły przez obliczanie koniunkcji i zastosowanie prawa pochłaniania dopóki jest to możliwe. Zakładając, że koniunkcja każdych dwóch dyzjunkcji o n komponentach daje n^2 komponentów (żadne komponenty nie ulegają redukcji) to złożoność obliczeniowa problemu redukcji p dyzjunkcji jest wykładnicza $O(n^p)$. Choć zazwyczaj podczas koniunkcji formuły ulegają redukcji to problem redukcji formuł boolowskich pozostaje złożonym problemem obliczeniowym. W związku z tym ważne jest zastosowanie odpowiednich strategii obliczeń jak i odpowiednia reprezentacja komputerowa problemu. W programie zastosowano dynamiczne struktury (listy i tablice) a zbiory atrybutów w komponencie są reprezentowane przez bitowe tablice. Reprezentacja taka pozwala na oszczędność pamięci i czasu wykonania. Koniunkcja bitów (realizowana przez

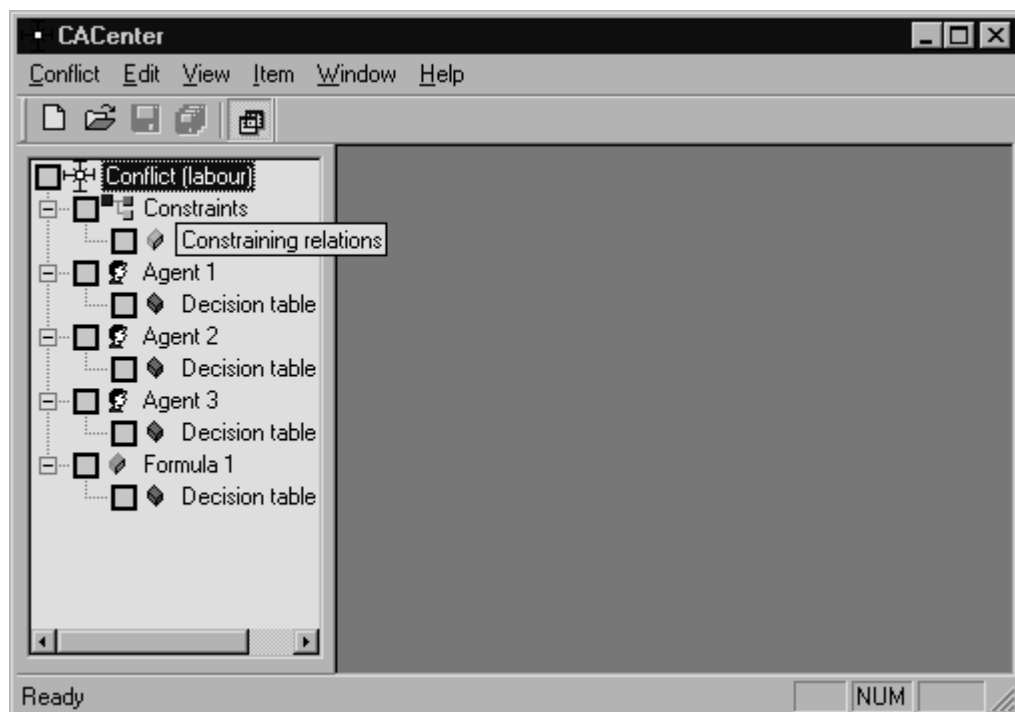
procesor) daje nam atrybuty występujące w obu komponentach. Porównanie wartości tych atrybutów prowadzi do redukcji lub utworzenia nowego komponentu (logiczna suma bitów).

9.2 Opis posługiwania się programem

Interfejs programu został zaprojektowany i zrealizowany w ten sposób aby posługiwanie się programem było intuicyjne dla użytkowników Windows. Do elementów ułatwiających posługiwanie się programem należą przede wszystkim widok projektu, menu kontekstowe, pasek narzędzi czy klawisze skrótów.

Główną ideą programu było umożliwienie przeprowadzenia różnorodnych analiz opartych na wnioskowaniu boolowskim w zaproponowanym modelu. Można więc dodawać nowe formuły, modyfikować je czy wybierać tylko część formuł do obliczeń.

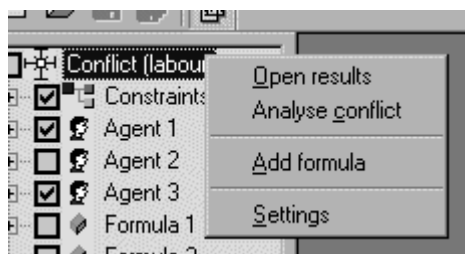
Pracę z programem rozpoczynamy od stworzenia projektu konfliktu – nadanie mu nazwy. Wszystkie pliki projektu są przechowywane w osobnym folderze, stworzonym na podstawie nazwy projektu.



Rysunek 2 Program analizy konfliktów (Conflict Analysis Center) widok po otwarciu projektu.

9.2.1 Widok projektu

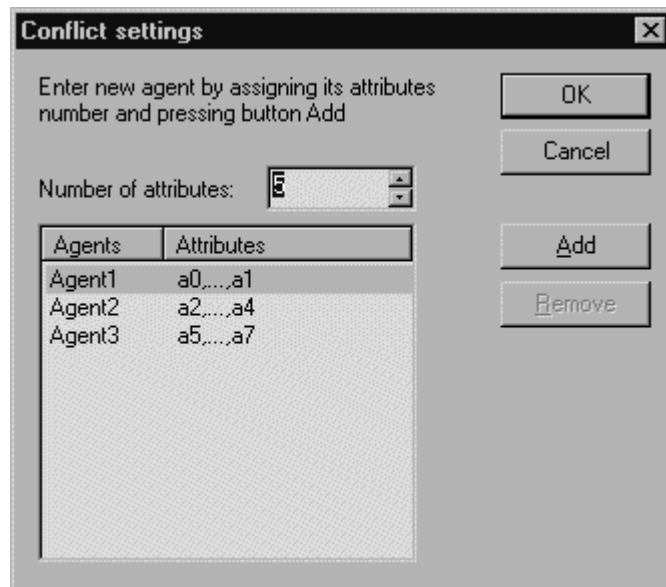
Informacje na temat projektu są przedstawione z pomocą drzewa (Rysunek 2). Korzeń drzewa symbolizuje konflikt. Liśćmi drzewa są różnego typu formuły boolowskie reprezentowane przez agentów biorących udział w konflikcie, ograniczenia oraz formuły dodatkowe. Z każdym agentem jak i formułą dodatkową może być związana tablica decyzyjna. Dla agentów odpowiada ona tablicy stanów lokalnych z subiektywną oceną. Tablice te mogą zostać przekształcone w formuły z pomocą polecenia *Calculate formula*. Z założenia opis klasy decyzyjnej jest wyznaczany dla wartości atrybutu decyzyjnego równej 1. Przy każdym liściu prócz ikony znajduje się pole zaznaczania (checkbox). Zaznaczona formuła jest brana pod uwagę podczas analizy konfliktu. Zaznaczając różne formuły można z pomocą jednego projektu wielostronnie analizować konflikt. Na przykład, zaznaczając tylko ograniczenia i dwóch agentów (Rysunek 3) można wyznaczyć pokojową przestrzeń ruchów dla trzeciego agenta. Zaznaczając natomiast formułę reprezentującą sytuacje dobre (Formuła 1) i ograniczenia (Constraints) i nie zaznaczając żadnych agentów analizujemy problem konsensusu na poziomie globalnym (punkt 5.1) itd.



Rysunek 3 Menu kontekstowe konfliktu.

Z każdym liściem drzewa jest związane odpowiednie menu kontekstowe, pozwalające na wykonanie dostępnych w programie operacji. Rysunek 3 przedstawia menu kontekstowe korzenia *Conflict*. Nowo utworzony projekt konfliktu nie zawiera żadnych agentów ani formuł. Liczbę agentów i ich atrybuty są ustawiane w okienku dialogowym wywoływanym poleceniem *Settings* (Rysunek 4). Atrybuty agentów są numerowane kolejno, poczynając od *a0* (pierwszy atrybut pierwszego agenta). Dodanie nowego agenta wymaga podania liczby, używanych przez agenta, atrybutów – program przypisze kolejne nazwy atrybutów. O użycie

odpowiednich atrybutów w formułach dodanych do projektu musi zadbać użytkownik. Do każdej formuły można przypisać tablicę decyzyjną.



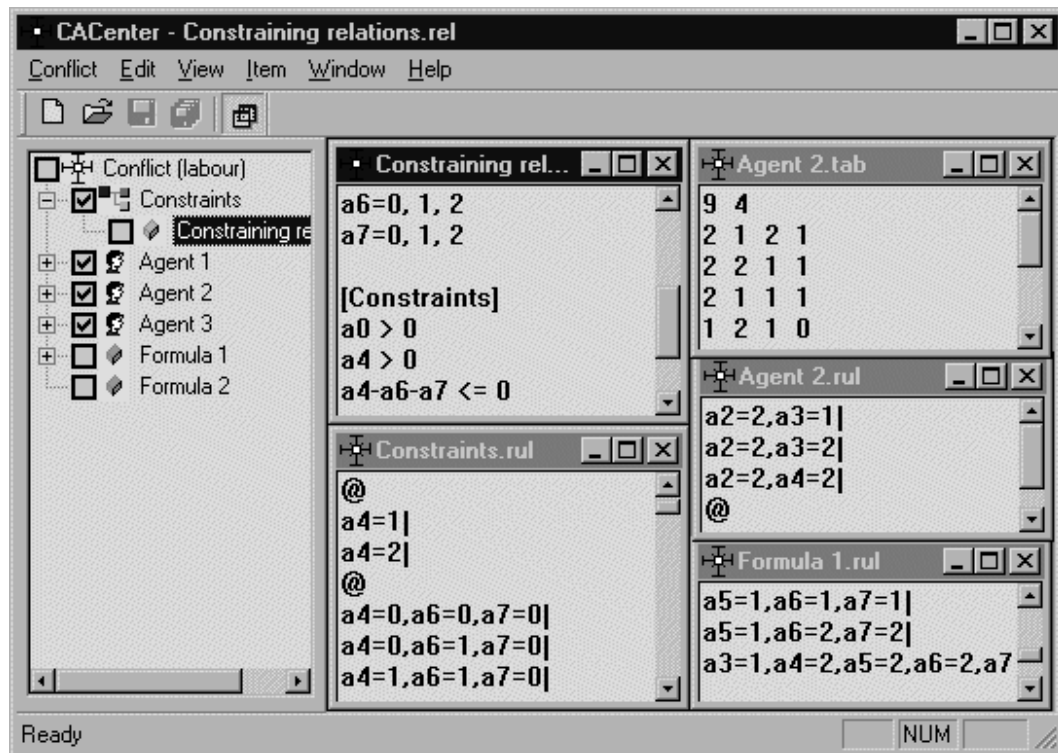
Rysunek 4 Dodawanie agentów do projektu konfliktu.

9.2.2 Pliki używane w programie

Jak wspomniano we wstępie wszystkie dane programu są przechowywane w plikach tekstowych. Prócz pliku opisującego projekt wszystkie pliki można tworzyć i modyfikować w programie. Pliki te są związane z każdym elementem drzewa projektu. Są one otwierane poleceniem *Edit* lub naciskając klawisz *Enter* na wybranym elemencie.

Wszystkie liście drzewa na pierwszym poziomie reprezentują formuły boolowskie: formuła opisująca ograniczenia, formuły opisujące zbiory celów agentów, formuły dodatkowe. Pliki przechowujące formuły mają nazwę zgodną z nazwą liścia i rozszerzenie ".rul". Koniunkcję deskryptorów oznacza znak , (przecinek) dyzjunkcję komponentów znak | (pionowa kreska), a koniunkcję dyzjunkcji oznaczono znakiem @ (at). Przykładową formułę można odnaleźć na widoku okna programu (Rysunek 5).

Tablice decyzyjne są przypisane do formuł i przechowywane w plikach o nazwach takich jak formuły lecz z rozszerzeniem ".tab". Format pliku opisującego tablicę decyzyjną przedstawiono w punkcie 9.1.



Rysunek 5 Przykłady plików używanych w programie analizy konfliktów.

Równości i nierówności opisujące ograniczenia są przechowywane w pliku z rozszerzeniem ".rel". Ma on postać pliku .ini, w którym wykorzystano dwie sekcje. Oprócz zależności między atrybutami, zapisanymi w sekcji *Constraints*, potrzebne są również dziedziny atrybutów użytych w ograniczeniach. W sekcji *Domains* przy kolejnych atrybutach wymienia się możliwe wartości atrybutu oddzielając je przecinkami (Rysunek 5). Informacja ta jest niezbędna do generacji formuły na podstawie ograniczeń (patrz punkt 9.1).

Wyniki analiz w postaci implikantów pierwszych są dopisywane do pliku wynikowego z rozszerzeniem ".out". Można go otworzyć poleceniem *Open results* z menu kontekstowego korzenia drzewa projektu.

9.2.3 *Analizy*

Przed przystąpieniem do analiz należy przygotować, wprowadzić dane do odpowiednich plików. Podstawowe dane w modelu są przedstawione w postaci tablic decyzyjnych; pochodzą one od agentów, od eksperta (sytuacje dobre) lub reprezentują dane historyczne (strategie). Wywołując polecenie *Calculate formula* wyznaczany jest minimalny, ze względu na liczbę użytych atrybutów, opis klasy decyzyjnej dla wartości atrybutu decyzyjnego równej 1. Wynik zapisywany jest w odpowiednim pliku, tak że można go modyfikować "otwierając" odpowiedni element drzewa (agent, formuła). W przypadku ograniczeń działanie programu jest takie samo, z tym że polecenie *Calculate formula* uruchamia modul programowania liniowego do wyznaczenia formuły.

Po przygotowaniu formuł analiza konfliktu sprowadza się do redukcji złożonej formuły boolowskiej. Formuły, które mają być brane pod uwagę podczas analizy należy zaznaczyć (pole checkbox) klikając myszką lub naciskając klawisz spacji. Jeśli na przykład formuła 1 zawiera opis sytuacji dobrych to zaznaczając tę formułę i ograniczenia (Constraints) analizujemy problem konsensusu na poziomie globalnym (patrz punkt 5.1). Jeśli dodatkowo zaznaczymy agentów to będziemy analizować problem konsensusu na poziomie globalnym i lokalnym (punkt 5.2) itd. Wybierając teraz polecenie *Analyse conflict* wykonywana jest redukcja koniunkcji zaznaczonych formuł. Wynik obliczeń można zobaczyć wybierając polecenie *Open results*.

9.2.4 *Analiza problemu konsensusu na poziomie globalnym i lokalnym*

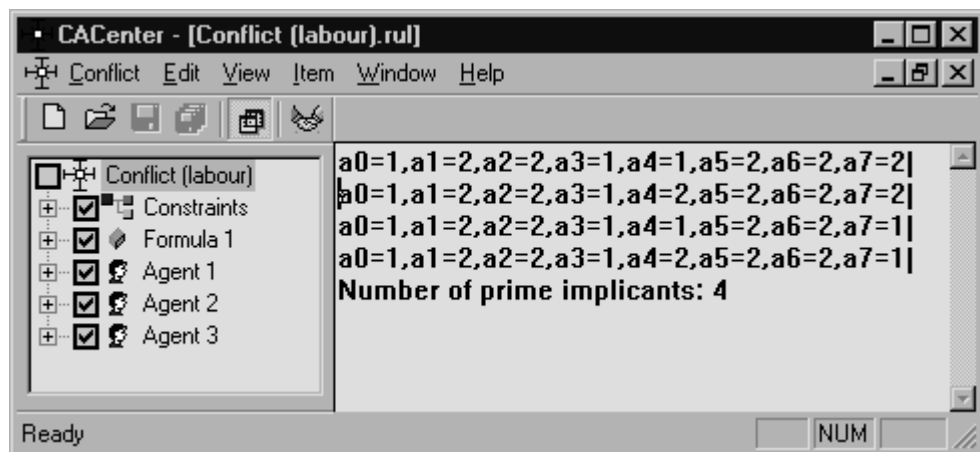
Przeanalizujmy przebieg eksperymentu polegającego na rozwiązaniu konfliktu między pracodawcą a pracownikami z pomocą programu wspomagania analizy konfliktów. Algorytm poszukiwania konsensusu na poziomie globalnym i lokalnym został opisany w punkcie 5.2. Aby przeprowadzić analizę z pomocą programu należy najpierw wprowadzić wszystkie niezbędne dane. Podstawowymi danymi identyfikującymi dany konflikt jest: nazwa konfliktu, liczba agentów biorących udział w konflikcie (wraz z liczbą atrybutów każdego agenta), ograniczenia oraz dodatkowe formuły. Wszystkie te informacje są zilustrowane w postaci drzewa konfliktu (Rysunek 2).

Niech nasz przykładowy konflikt nazywa się „labour” – co wpisujemy w okienko dialogowe po wybraniu opcji menu *Conflict/New*. Początkowo drzewo konfliktu będzie składało się z jednej tylko gałęzi – *Constraints* zawierającej jeden liść *Constraining relations*. Następnie wybieramy polecenie menu *Conflict/Settings* gdzie wprowadzamy liczbę atrybutów dla każdego agenta – odpowiednio 2 dla pierwszego i 3 dla drugiego i trzeciego. Rysunek 4 przedstawia okienko dialogowe po wprowadzeniu informacji o wszystkich agentach dla naszego konfliktu. W drzewie konfliktu powinny pojawić się teraz trzy dodatkowe gałęzie oznaczające wprowadzonych agentów. Ponieważ do analizy konfliktu potrzebujemy jeszcze formułę opisującą sytuacje dobre z menu kontekstowego dodajemy nową formułę do drzewa konfliktu (polecenie *Add formula*). Formuły opisujące stanowiska agentów (zbiory celów) jak i formułę opisującą sytuacje dobre wyznaczamy na podstawie tablic decyzyjnych. Dodajemy więc do gałęzi drzewa liście oznaczające tablice decyzyjne dla agentów i formuły 1. (Rysunek 2 przedstawia stan drzewa konfliktu dla omawianego eksperymentu). Teraz można już wprowadzić tablice decyzyjne do plików (dostępnych z odpowiednich liści drzewa). Tablice decyzyjne wprowadzamy w odpowiednim formacie (patrz 9.2.2). I tak np. tablicę decyzyjną sytuacji ocenionych przez eksperta (Tabela 12) należy poprzedzić nagłówkiem "28 9" ponieważ zawiera 28 wierszy i 9 kolumn. Pozostaje nam wprowadzenie jeszcze jednej informacji definiującej konflikt – ograniczeń. Ograniczenia wprowadzamy w pliku skojarzonym z liściem "Constraining relations" w formacie opisanym w punkcie 9.2.2.

Po wprowadzeniu wszystkich niezbędnych informacji możemy przejść do obliczeń. Pierwszym etapem jest przekształcenie ograniczeń do postaci formuły boolowskiej. Wywołujemy więc polecenie *Calculate formula* z menu kontekstowego liścia "Constraining relations". Wynik działania procedury możemy zobaczyć otwierając plik skojarzony z gałęzią drzewa "Constraints". Formuła jest dość długa – jest koniunkcją sześciu podformuł (w przykładowym konflikcie jest sześć ograniczeń 3.3.1). Następnie dla każdej tablicy decyzyjnej wyznaczymy opis klasy decyzyjnej dla decyzji 1. Wyznaczone formuły będą odpowiednio opisywać stany preferowane przez agentów (ich zbiory celów) oraz sytuacje obiektywnie dobre. W programie dla każdego liścia "Decision table" wywołujemy polecenie *Calculate formula* z menu kontekstowego. Przykłady różnych plików używanych w analizie konfliktu między pracodawcą i pracownikami przedstawia Rysunek 5. Dokładniej okienka zawierające dane z plików "Constraining relations.rel" i "Constraints.rul" odpowiadają odpowiednio

opisowi ograniczeń w postaci nierówności oraz formule boolowskiej, do której te ograniczenia zostały przekształcone. Następne pliki "Agent 2.tab" i "Agent 2.rul" przedstawiają odpowiednio tablicę decyzyjną agenta drugiego i opis klasy decyzyjnej dla decyzji 1. Ostatni widoczny na rysunku plik ("Formuła 1.rul") przedstawia fragment formuły opisującej sytuację dobre.

W celu rozwiązania konfliktu – znalezienia sytuacji niekonfliktowych na poziomie globalnym i lokalnym należy zaznaczyć wszystkie gałęzie w drzewie konfliktu (reprezentujące agentów 1 do 3, ograniczenia i formułę 1). Następnie z menu *Conflict* należy wybrać polecenie *Analyse conflict*. Wynik działania procedury można obejrzeć np. poprzez wywołanie polecenia *Open results* z menu kontekstowego (Rysunek 6). Otrzymany wynik jest zgodnym z tym przedstawionym w punkcie 5.2.



Rysunek 6 Wynik analizy problemu konsensusu na poziomie globalnym i lokalnym

WNIOSKI

W pracy zaproponowano model oparty o teorię zbiorów przybliżonych. Jest on rozwinięciem prostego modelu Pawłaka. Poprzez wprowadzenie stanów lokalnych agentów i ograniczeń zaproponowany model daje możliwość identyfikacji przyczyn konfliktu. Uwzględnienie preferencji agentów i/lub dobra wspólnego jest podstawą poszukiwania konsensusu i rozwiązania konfliktu. Mimo prostego opisu sytuacji konfliktowej w zaproponowanym modelu można analizować wiele problemów w tym również związanych z ich dynamiką. Zastosowanie teorii zbiorów przybliżonych pozwoliło przezwyciężyć problem niekompletności danych tak ważny przy analizie sytuacji konfliktowych. W połączeniu z wnioskowaniem boolowskim teoria zbiorów przybliżonych wprowadza uniwersalną, silną i prostą metodę analizy. Punktem wyjścia są tablice informacyjne przekształcane w tablice decyzyjne. Utworzone na ich podstawie reguły decyzyjne aproksymują opis decyzji (wyłącznie na podstawie dostępnej informacji). Pozostaje tylko zakodować problem w postaci formuły boolowskiej tak, aby implikanty pierwsze formuły dawały rozwiązania problemu. Przedstawiona powyżej metoda jest podstawą analiz przedstawionych w pracy – na przykład problemu konsensusu. Metoda ta prowadzi też do prostej implementacji komputerowej i opiera się na dwóch modułach. Pierwszym z nich jest moduł generacji opisu klasy decyzyjnej dla wybranej decyzji z tablicy decyzyjnej (lewe strony minimalnych reguł decyzyjnych), zadaniem drugiego modułu jest redukcja formuły boolowskiej do postaci minimalnej dyzjunkcji. Oba te moduły są najważniejszą częścią programu przygotowanego przez autora i towarzyszącego pracy. Wspomaganie komputerowe analizy konfliktów jest niezbędne przy zastosowanej metodzie. Wnioskowanie boolowskie wymaga bowiem operowania na złożonych formułach boolowskich.

Podsumowując najważniejszymi zaletami przedstawionego modelu są:

- zamiast posługiwania się modelami analitycznymi, trudnymi do odgadnięcia zwłaszcza przy dużej liczbie agentów, konflikty modeluje się w oparciu o dane

eksperymentalne z zastosowaniem wnioskowania aproksymacyjnego pozwalającego uzyskać aproksymacyjny opis analizowanych pojęć,

- ogólność modelu (możliwość modelowania różnych typów konfliktów),
- możliwość uwzględnienia dowolnej liczby agentów biorących udział w konflikcie,
- bezpośredni opis sytuacji konfliktowych za pomocą rzeczywistych danych (stany lokalne),
- aproksymacja niepełnej wiedzy w procesie syntezy reguł decyzyjnych,
- wprowadzenie ograniczeń, wskazanie przyczyn konfliktów,
- możliwość monitorowania, przewidywania konfliktu,
- uwzględnienie dobra wspólnego w poszukiwaniu konsensusu,
- użycie wnioskowania boolowskiego do analizy konfliktu,
- czytelny opis sytuacji konfliktowej,
- łatwość implementacji komputerowej.

Poważnym ograniczeniem zaproponowanego modelu może być złożoność obliczeniowa algorytmów związanych z manipulowaniem dużymi funkcjami boolowskimi – co w praktyce oznacza konieczność zastosowania zaawansowanych heurystyk pozwalających wyznaczyć rozwiązania suboptymalne. Innym problemem, przy modelowaniu rzeczywistego konfliktu może być identyfikacja ograniczeń (więzów) występujących w danej sytuacji.

10.1 Kierunki dalszych badań

Najbardziej podstawowym kierunkiem dalszych badań w zaproponowanym modelu analizy konfliktów jest opracowanie dobrych algorytmów aproksymacyjnych do redukcji formuł boolowskich. Opracowanie takich algorytmów wydaje się niezbędnym warunkiem analizy rzeczywistych konfliktów charakteryzujących się dużą liczbą danych (duża liczba agentów, mnogość atrybutów i ich zbiorów wartości, duża liczba ograniczeń). Opracowanie skutecznych heurystyk dla ogólnego problemu redukcji formuł boolowskich nie wydaje się być szybko osiągalne [59]. Dlatego warto rozważyć algorytmy aproksymacyjne opierające się na informacjach, które dodatkowo można uzyskać w analizie konfliktów (np. znajomość koalicji agentów).

Wracając do modelu Pawlaka, w różnych sytuacjach konfliktowych dochodzi do rozstrzygnięć z pomocą głosowania. Agenci wyrażają wtedy swoją opinię na pewne wybrane sprawy (*sprzeciw, neutralność, aprobatę*) – odpowiada to opisowi konfliktu w modelu Pawlaka (patrz Tabela 1). Wydaje się więc oczywistym, że powinna istnieć zależność między sprawami zaproponowanymi do głosowania i sprawami opisującymi konflikt w nowym modelu. Decyzje agentów podjęte w takim głosowaniu muszą odzwierciedlać preferencje agentów zapisane poprzez ich zbiory celów. Co więcej na podstawie danych z nowego modelu można by było zweryfikować słuszność wyboru atrybutów poddanych pod głosowanie. Zaproponowanie takich atrybutów na podstawie informacji zawartych w nowym modelu pozostaje problemem na razie nierozwiązanym.

Wiele problemów decyzyjnych jest związanych z uzgodnieniem stanowiska przez niezależnych agentów (handel w sieci komputerowej, współpraca podczas produkcji, uzgadnianie terminów itd.). W zaproponowanym modelu można poszukiwać konsensusu – wyznaczać sytuacje możliwe do zaakceptowania przez wszystkich agentów (punkt 5.3). Często jednak istnieje potrzeba negocjacji stanowisk zwłaszcza, gdy poszukiwanie konsensusu przy wyspecyfikowanych celach agentów nie powiodło się. Negocjacje mogą być w opisanym modelu przedstawione jako proces modyfikowania zbiorów celów agentów o nowe stany do momentu uzyskania konsensusu.

ODNIESIENIE DO TEORII GIER

Jak wspomniano we wstępie teoria gier stanowi podstawę wielu prac poświęconych analizie konfliktów i praktycznych zastosowań. Warto tu wspomnieć, że niestety najbardziej rozwijającą się gałęzią zastosowań są symulacje działań bojowych i gry wojenne [36]. Podczas działań wojennych w Zatoce Perskiej wykorzystano w procesie podejmowania decyzji np. system "Janus" oparty na stochastycznej, dwustronnej grze wojennej.

Zaproponowany w pracy model mimo wielu zbieżności definiuje sytuację konfliktową w odmienny sposób. Całkowicie też różne są zastosowane metody analizy konfliktów. Choć pozwala na wprowadzenie elementów gry, sam konflikt nie jest rozpatrywany jako gra (maksymalizacja zysku) lecz jako pewne zadanie optymalizacyjne z ograniczeniami. Ważnym elementem zaproponowanego modelu jest pojęcie dobra wspólnego, które z zasady nie występuje w teorii gier. (Koopercja w teorii gier bardziej dotyczy podziału zysku a nie polega na uwzględnieniu nadrzędnego dobra wspólnego). W tym rozdziale spróbujemy przyjrzeć się podstawowym podobieństwom i różnicom zaproponowanego modelu i modelom w teorii gier. Teoria gier jest dziedziną rozwijającą się od wielu lat, w rozdziale tym odniosę się jedynie do najważniejszych typów gier z punktu widzenia analizy konfliktów.

11.1 Podstawowe elementy teorii gier

Strategia w teorii gier jest dokładnym opisem metody zachowania się gracza w każdej możliwej sytuacji. *Sytuacją* (ang. outcome) jest każdy element iloczynu kartezyjskiego zbiorów strategii wszystkich graczy. W klasycznej teorii gier każdej sytuacji jest przypisana *wartość wypłaty* dla każdego gracza (ang. payoff). W rzeczywistych sytuacjach konfliktowych takie podejście jest w oczywisty sposób niewygodne, dlatego powstała teoria użyteczności (ang. utility theory), w myśl której do zbioru sytuacji wprowadza się porządek dla każdego gracza. Dokładniej wyższa *wartość użyteczności* dla pewnego gracza oznacza sytuację bardziej przez niego preferowaną. Celem każdego gracza w grze jest maksymalizacja własnej

użyteczności. Sytuacją *korzystną* (ang. rational) dla gracza nazywamy sytuację o największej wartości użyteczności tego gracza przy niezmienionych strategiach innych graczy. Sytuacja korzystna dla wszystkich graczy nazywana jest sytuacją *równowagi* (ang. equilibrium). Tradycyjnie w teorii gier rozwiązaniem konfliktu jest wyznaczenie wszystkich sytuacji równowagi.

Nie wnikając w szczegóły (różne modele gier) można znaleźć odniesienie definiowanych pojęć w teorii gier do pojęć prezentowanych w pracy. Można je zestawić w poniższej tabeli:

model w teorii gier	model oparty o teorię zbiorów przybliżonych
zbiór strategii wybranego gracza i: S_i	zbiór stanów lokalnych agenta $ag_i: U_{ag_i}$
zbiór sytuacji: $Q=S_1 \times \dots \times S_n$ (n graczy)	zbiór sytuacji osiągalnych: $U_{ag_1} \times \dots \times U_{ag_n}$ (n agentów), zawężony przez ograniczenia
użyteczność (funkcja preferencji)	preferencje agentów przeniesione na poziom sytuacji
rozwiązanie konfliktu: znalezienie punktu równowagi	konsensus na poziomie globalnym (ocena agentów)

Tabela 21 Porównanie podstawowych elementów modelu zdefiniowanego w pracy i modelu opartego o teorię gier.

Mimo tych wydawałoby się zbieżności w definiowaniu sytuacji konfliktowych podejście zaprezentowane w pracy różni się o tego z teorii gier. Podstawą analizy w zaproponowanym modelu są rzeczywiste dane opisujące stany lokalne w postaci tablic informacyjnych. Odpowiadające stanom lokalnym strategie w teorii gier (Tabela 21) nie opisują stanów agentów, ale są opisem zachowania się agenta w każdej możliwej sytuacji. (Dlatego w pracy zdefiniowano pojęcie strategii niezależnie od pojęcia stanów lokalnych). W rzeczywistych konfliktach liczba sytuacji może być oczywiście bardzo duża, a rozważanie każdej z osobna czasochłonne. Wątpliwa jest również możliwość wyznaczenia wszystkich strategii danego agenta. Teoria zbiorów przybliżonych pozwala przeprowadzać proces wnioskowania przy niepełnej i niepewnej informacji. W sytuacjach konfliktowych pozwala również na szukanie rozwiązań wśród sytuacji nie rozważanych przez agentów.

W pracy wskazano ograniczenia zasobów jako fundamentalny powód powstawania konfliktów. W teorii gier powody konfliktów nie są jasne (co jest powodem sprzeczności dążeń?). Ze względu na ograniczenia i uwzględnienie niepewności informacji najbardziej zbliżonym typem gier, do modelu zastosowanego w pracy, są gry w warunkach niepewności [33]. W grach tych jeden z graczy reprezentuje naturę. Ze względu na brak strategii natury wprowadza się zmienne losowe, których rozkład zależy od stanu natury. Na podstawie obserwacji tych zmiennych każdy z graczy przyjmuje strategię. Podobnie dla ustalonego stanu natury wartość funkcji wypłaty każdego gracza jest zmienną losową. Oczywiście istnieje tu problem wyznaczenia rozwiązania tak opisanego gry. Trudności wiążą się z określeniem rozkładu zmiennych losowych oraz zmian stanów natury. Zaproponowany w pracy opis zasobów jak i metody obliczeniowe mogą okazać się skuteczniejsze przy analizie złożonych konfliktów, ponadto metody te opierają się na danych eksperymentalnych.

Przykład 11.1

Zobaczmy na przykładzie możliwego konfliktu nuklearnego między Stanami Zjednoczonymi (U.S.) i Związkiem Radzieckim (USSR) sposoby analizy konfliktu w Teorii Gier (opis konfliktu zaczerpnięty z [19]). Konflikt ten zostanie zamodelowany i analizowany na trzy sposoby: za pomocą klasycznej teorii gier, za pomocą metagry Howarda i za pomocą analizy konfliktów zaproponowanej przez Fraser i Hipel. Niech będą dane następujące strategie dla obu graczy C – atak konwencjonalny, L – ograniczony atak nuklearny i S – pełny atak nuklearny. Grę w postaci normalnej przedstawia Tabela 22 (liczby w tabeli oznaczają wartości funkcji wypłaty dla graczy w danej strategii).

U.S. (gracz 1)	USSR (gracz 2)		
	C	L	S
C	5, 8	R_2 4, 9	1, 7
L	R_1 9, 4	R_2 6, 6	2, 5
S	8, 1	R_1 7, 2	R_1R_2 3, 3

Tabela 22 Postać normalna konfliktu między US i USSR.

Sytuacją korzystną dla obu graczy jest sytuacja (S, S) , która jest punktem równowagi i rozwiązaniem gry. Trudno pogodzić się z wynikiem, że obustronny pełny atak nuklearny jest rozwiązaniem konfliktu – sytuacja ta gwarantuje jedynie, że (teoretycznie) żaden gracz nie przegrywa. W przypadku tego konfliktu bardziej interesowałyby nas możliwe rozwiązania kooperacyjne. Łatwo zauważyć, że np. sytuacja (L, L) daje poprawę, zwiększa wartość wypłaty, dla obu graczy. Podobnie kooperacyjnymi rozwiązaniami konfliktu są sytuacje (C, C) , (C, L) i (L, C) , które są logicznie bliższe do zaakceptowania niż niszcząca świat totalna wojna nuklearna. Postać normalna gry nie nadaje się do analizy konfliktów z większą liczbą graczy.

11.2 Metagra Howarda

Metagra jest grą, w której bierze się pod uwagę możliwe reakcje danego gracza na znane strategie innych graczy. Idea brania pod uwagę reakcji graczy na reakcję graczy prowadzi do utworzenia hierarchii metagier [24] opisanych przez nieskończone drzewo Howarda. Aby można było prowadzić analizę konfliktów, mimo nieskończoności drzewa Howard wyprowadził Twierdzenie Charakterystyczne. Wynika z niego, że można analizować wszystkie metagry, na podstawie gry wyjściowej. Sytuacje, które są racjonalne w danej matagrze są metaracjonalne w grze wyjściowej. Pojęcie metaracjonalności jest podstawą do analizy stabilności i poszukiwania rozwiązania konfliktu. Rozróżnia się trzy rodzaje stabilności: racjonalność, symetryczną metaracjonalność i generalną metaracjonalność. Analizę w tym modelu dokonuje się na podstawie tablicy, w której są ujęci wszyscy gracze i wszystkie możliwe sytuacje (iloczyn kartezjański strategii graczy). Wybrana strategia danego gracza jest zaznaczona cyfrą jeden, przy pozostałych stawia się zero. Tabela 23 prezentuje opis konfliktu nuklearnego między U.S. a USSR za pomocą metagry.

	Sytuacje								
U.S.									
<i>C</i>	1	1	1	0	0	0	0	0	0
<i>L</i>	0	0	0	1	1	1	0	0	0
<i>S</i>	0	0	0	0	0	0	1	1	1

Sytuacje									
USSR									
<i>C</i>	0	1	0	0	0	1	0	0	1
<i>L</i>	1	0	0	1	0	0	0	1	0
<i>S</i>	0	0	1	0	1	0	1	0	0

Tabela 23 Metagra – konflikt między US a USSR.

Można zauważyć, że wszystkich możliwych sytuacji dla n graczy będzie n^m gdzie m jest liczbą strategii w danym konflikcie. Bardzo ważnym, oddzielnym problemem jest więc redukcja sytuacji logicznie niedopuszczalnych.

Każdą sytuację należy teraz przeanalizować z punktu widzenia każdego gracza (analiza stabilności). Dla każdego gracza wyznaczmy sytuacje *racjonalne*, *symetrycznie metaracjonalne* i *generalnie metaracjonalne*, pozostałe sytuacje są *niestabilne*. Sytuacje przedstawione w tabeli (Tabela 23) są ustawione zgodnie z preferencjami USSR – od najbardziej do najmniej preferowanych.

Sytuacje *racjonalne* są to takie sytuacje, których agent nie jest w stanie zmienić na korzystniejsze przez wybór innej strategii – czyli poprzez UI (ang. unilateral improvement). Na przykład USSR sytuację (010, 100) może poprawić przechodząc do bardziej preferowanych sytuacji (010, 001) lub (010, 010). Natomiast sytuacja (010, 010) jest racjonalna dla USSR – gracz nie może jej już poprawić.

Sytuacja jest *symetrycznie racjonalna* dla danego gracza, gdy dla każdego UI w tej sytuacji, inny gracz lub gracze mają taką strategię, która powoduje znalezienie się gracza w mniej preferowanej sytuacji. Strategia pozostałych graczy jest w takim przypadku nazywana *strategią sankcjonowania* (ang. sanctioning strategy). Przykładem takiej sytuacji jest sytuacja (100, 100) dla USSR. Gracz ten może poprawić sytuację przechodząc do (100, 010), co jednak spowoduje (przez UI dla U.S.) przejście do sytuacji (010, 010) lub (001, 010) mniej korzystnych dla USSR.

Generalnie metaracjonalna sytuacja występuje wtedy, gdy dany gracz ma w tej sytuacji UI ale inni gracze mogą wybrać strategię, która spowoduje znalezienie się gracza w mniej preferowanej sytuacji niż wyjściowa. Jednak dany gracz posiada w nowej sytuacji strategię,

która znowu może poprawić sytuację gracza. Ten cykl może trwać nieskończenie długo, dlatego wydaje się, że gracz pozostanie w wyjściowej sytuacji. W przykładowym konflikcie nie występują sytuacje generalnie metaracjonalne.

Przykład 11.2

Tabela 24 prezentuje wyniki analizy wszystkich sytuacji w konflikcie między U.S i USSR.

Sytuacja	Stabilność		
	U.S.	USSR	konflikt
(100, 100)	symetrycznie metaracjonalna	symetrycznie metaracjonalna	punkt równowagi (jeśli wiarygodna)
(100, 010)	symetrycznie metaracjonalna	racjonalna	punkt równowagi (jeśli wiarygodna)
(100, 001)	niestabilna	symetrycznie metaracjonalna	niestabilna
(010, 100)	racjonalna	symetrycznie metaracjonalna	punkt równowagi (jeśli wiarygodna)
(010, 010)	symetrycznie metaracjonalna	racjonalna	punkt równowagi (jeśli wiarygodna)
(010, 001)	niestabilna	symetrycznie metaracjonalna	niestabilna
(001, 100)	symetrycznie metaracjonalna	niestabilna	niestabilna
(001, 010)	racjonalna	niestabilna	niestabilna
(001, 001)	racjonalna	racjonalna	punkt równowagi

Tabela 24 Wyniki analizy metagry w konflikcie między US a USSR.

Punkty równowagi wynikające z symetrycznej i generalnej metaracjonalności należy poddać jeszcze analizie wiarygodności (ang. credibility). Należy rozważyć realność zastosowania sankcji odpowiedzialnych za stabilność danej sytuacji. Jeżeli sankcje w danej sytuacji nie są realne należy uznać rozważaną sytuację za niestabilną. Ścisły algorytm analizy wiarygodności nie został podany – co ogranicza wykorzystanie metagry w analizie konfliktów z pomocą komputera.

11.3 Analiza konfliktów (wg Fraser i Hipel [19])

Wprowadzając założenie, że wiarygodność sankcji jest zależna od preferencji agentów Fraser i Hipel około roku 1984 [19] zaproponowali metodę analizy konfliktów opartą na metagrze Howarda. Punktem wyjścia jest wyznaczenie wektorów preferencji sytuacji dla każdego gracza – uszeregowanie sytuacji od najbardziej do najmniej korzystnych. Dla wygody analizy (szczególnie sprawdzenia występowania *symulacyjnego sankcjonowania*) sytuację przedstawia się w postaci liczb dziesiętnych – w zapisie z metagry (Tabela 23) każda strategia jest traktowana jako pozycja w zapisie dwójkowym np. sytuacja (100, 010) będzie teraz oznaczona liczbą 17. Prócz sytuacji racjonalnych i niestabilnych Fraser i Hipel rozważają *sekwencyjne sankcjonowanie* i *symulacyjne sankcjonowanie* jako podstawę stabilności sytuacji. *Sekwencyjne sankcjonowanie* występuje wtedy, gdy dla wszystkich UI w rozważanej sytuacji dla danego gracza, wiarygodna akcja może zostać podjęta przez innych graczy, tak że spowoduje ona znalezienie się rozważanego agenta w sytuacji mniej preferowanej niż wyjściowa. Akcją wiarygodną nazywamy taką akcję, w wyniku której gracz, który ją podjął znajdzie się w sytuacji bardziej preferowanej. Możliwość znalezienia się w gorszej sytuacji niż wyjściowa powstrzymuje gracza od jednostronnej poprawy (UI). *Symulacyjne sankcjonowanie* natomiast występuje wtedy, gdy wszyscy gracze jednocześnie zmieniają strategię z sytuacji niestabilnej lecz w efekcie wszyscy lub część z nich znajdzie się w sytuacji mniej korzystnej. Ostatecznie więc wszyscy lub część graczy zrezygnuje ze zmiany strategii – dla nich rozważana sytuacja jest stabilna. Algorytm analizy stabilności według Fraser i Hipel można przedstawić w następujących krokach.

1. Skonstruuj tablicę opisującą konflikt:
 - a. Wyznacz wektor preferencji dla każdego gracza (sytuacje oznaczone liczbami dziesiętnymi).
 - b. Wyznacz wszystkie UI (jednostronna poprawa) dla każdej sytuacji (umieść je pod każdą sytuacją).
 - c. Zaznacz identycznie preferowane sytuacje.
2. Analizuj każdą sytuację pod względem stabilności:
 - a. Zaznacz jako racjonalne (r) wszystkie sytuacje, które nie mają UI.
 - b. Dla każdej sytuacji sprawdź czy jest niestabilna (u) lub czy jest sekwencyjnie

stabilna (s) analizując jej UI.

c. Jeśli sytuacja jest niestabilna sprawdź występowanie symulacyjnej stabilności.

d. Sytuacje stabilne dla wszystkich graczy stanowią punkty równowagi (E) w konflikcie (ang. equilibriums).

Przykład 11.3

Kompletne, dziesiętne wektory preferencji dla graczy w omawianym konflikcie są następujące [19]: dla U.S. [10 12 20 18 9 17 36 34 33] i dla USSR [17 9 33 18 34 10 36 20 12]. Tabela 25 przedstawia analizę konfliktu nuklearnego między U.S. a USSR – jest ona posortowana zgodnie z wektorem preferencji od najbardziej do najmniej preferowanej sytuacji – górna część dla U.S. a dolna dla USSR.

	E			E			E			
	r	s		r	s	s	u	r	u	u
U.S.	10	12	20	18	9	17	36	34	33	
		10		20	10	20		36	36	
					12	18				34
	r	s	s	r	s	s	r	u	u	
USSR	17	9	33	18	34	10	36	20	12	
		17	17		18	18		36	36	
			9			34				20

Tabela 25 Tablica analizy stabilności konfliktu między U.S. i USSR.

Należy teraz dokonać analizy stabilności dla każdej sytuacji. Rozważmy dla przykładu sytuację 12 dla U.S. Istnieje możliwość poprawy tej sytuacji (UI) – przejście do sytuacji 10. W tej sytuacji (oznaczonej przez 10) USSR może wykonać UI do sytuacji 18 lub 34. Obie te sytuacje są mniej preferowane przez U.S. niż sytuacja 12. U.S. nie wykona więc zmiany sytuacji i sytuacja 12 jest sekwencyjnie stabilna (s).

11.3.1 Porównanie wyników analizy stabilności

W teorii gier powstało wiele sposobów poszukiwania sytuacji równowagi. Należą do nich:

- Równowaga Nasha (sytuacje racjonalne dla wszystkich graczy) [32].

- Równowaga w metagrze (analiza stabilności w metagrze – patrz punkt 11.2) [24].
- Równowaga dalekowzroczna (Nonmyopic equilibriums) [8]. (Zdefiniowana tylko dla gier dwuosobowych przy założeniu, że każdy gracz ma dwie strategie. Jest wyznaczana poprzez drzewo gry, zakładając, że obaj gracze kolejno zmieniają strategię. Gracze zmieniają strategię, jeśli zmiana spowoduje ostatecznie osiągnięcie korzystniejszej sytuacji. Do tego ostatniego stosuje się metodę wstecznej indukcji)
- Stabilność Stackelberga [19] (W dwuosobowej grze zakłada się istnienie lidera i następcy. Lider optymalizuje wybór strategii na podstawie najlepszej odpowiedzi następcy. Przy czym następca odpowiada w taki sposób by znaleźć się w najlepszej sytuacji. Najlepszą sytuację, w której może znaleźć się lider nazywamy sytuacją równowagi Stackelberga.)

Zbiór sytuacji równowagi wyznaczony w analizie konfliktów zaproponowanej przez Fraser i Hipel zawiera wszystkie racjonalne sytuacje równowagi zdefiniowane przez wyżej wymienione typy stabilności (formalny dowód można znaleźć w [19]). Przykład 11.4 zawiera porównanie wyników zastosowania omawianych powyżej trzech typów analizy stabilności dla konfliktu nuklearnego.

Przykład 11.4

Analizując wyniki otrzymane za pomocą trzech typów analizy stabilności (Tabela 26, strona 97) można zauważyć, że są one identyczne za wyjątkiem interpretacji sytuacji 17. Tylko analiza konfliktów zaproponowana przez Fraser i Hipel nie interpretuje tej sytuacji jako punkt równowagi. Sytuacja ta, gdy USSR atakuje ograniczonymi siłami nuklearnymi, a U.S. odpowiada atakiem konwencjonalnym jest stabilna dla U.S. i niestabilna dla USSR (Tabela 25). Jeśli U.S. wykona UI do sytuacji 20 (pełny atak nuklearny) to USSR sankcjonuje to posunięcie przechodząc do sytuacji 36 (sytuacja 36 jest mniej preferowana przez U.S. niż 17). Jeżeli jednak U.S. wykona UI do sytuacji 18 USSR nie może poprawić swojej sytuacji – nie posiada wiarygodnego sankcjonowania. W konsekwencji sytuacja 17 jest sytuacją niestabilną dla U.S. Prawidłowe rozpoznawanie tego, kiedy sytuacja jest wiarygodnie sankcjonowana jest zaletą metody Fraser i Hipela w stosunku do innych typów analizy stabilności.

Sytuacja	Stabilność		
	Postać normalna	Analiza w metagrze	Analiza konfliktów Fraser i Hipel
9 (C, C) (100, 100)	rozwiązanie kooperacyjne	punkt równowagi	punkt równowagi
10 (L, C) (010, 100)	rozwiązanie kooperacyjne	punkt równowagi	punkt równowagi
12 (S, C) (001, 100)	–	–	–
17 (C, L) (100, 010)	rozwiązanie kooperacyjne	punkt równowagi	–
18 (L, L) (010, 010)	rozwiązanie kooperacyjne	punkt równowagi	punkt równowagi
20 (S, L) (001, 010)	–	–	–
33 (C, S) (100, 001)	–	–	–
34 (L, S) (010, 001)	–	–	–
36 (S, S) (001, 001)	punkt równowagi	punkt równowagi	punkt równowagi

Tabela 26 Wyniki trzech typów analizy stabilności konfliktu między US a USSR.

11.4 Podsumowanie

Można zauważyć pewne podobieństwa metagry Howarda i zastosowanego w pracy modelu (zwłaszcza w części poświęconej zmienności sytuacji konfliktowych i strategiom). Najważniejszym podobieństwem jest wprowadzenie w pracy grafu zmienności sytuacji konfliktowej, któremu odpowiada drzewo Howarda. W metagrze można też znaleźć odpowiedniki do przestrzeni rozwiązań i przestrzeni ruchów. Ponadto w obu przypadkach wykorzystuje się wektor preferencji danego agenta w analizie sytuacji konfliktowej.

Zasadniczą jednak różnicą jest sposób modelowania konfliktu i interpretacja rozwiązania. W zaproponowanym w pracy modelu nie interesuje nas partykularnie najkorzystniejsza odpowiedź danego agenta na strategię innego lecz zakładamy, że wszyscy agenci dążą do kompromisu (oczywiście w ramach możliwych do zaakceptowania celów). Dlatego rozwiązania konfliktu w zaproponowanym modelu są bliższe rozwiązaniom kooperacyjnym w teorii gier (osobno jest rozważana trwałość sytuacji). Inną znaczącą różnicą jest uwzględnienie w naszym modelu dobra wspólnego, które ogranicza możliwe sytuacje do sytuacji dobrych – sytuacja (S, S) z pewnością do nich nie należy.

Metody analizy konfliktów używane w teorii gier różnią się też zdecydowanie od zastosowanej w pracy metody wnioskowania boolowskiego. Jeżeli wybrany problem można zakodować w postaci formuły boolowskiej to rozwiązanie problemu polega na obliczeniu implikantów pierwszych tej formuły. Użycie wnioskowania boolowskiego daje możliwość przejrzystego formułowania problemów i rozwiązywania ich w standardowy sposób. Algorytm analizy stabilności wymaga przeanalizowania każdej sytuacji z osobna, a liczba sytuacji w rzeczywistych konfliktach może być bardzo duża. Jak wcześniej wspomniałem liczba sytuacji do przeanalizowania rośnie wykładniczo wraz ze wzrostem liczby strategii. Bardzo ważnym problemem przy zastosowaniu metod opartych na metagrze jest więc usuwanie zbędnych, logicznie niedopuszczalnych sytuacji – wykonanie tych redukcji bez udziału człowieka nie jest na razie w pełnym stopniu osiągalne.

PODSTAWY TEORII ZBIORÓW PRZYBLIŻONYCH I WNIOSKOWANIA BOOLOWSKIEGO

Teoria zbiorów przybliżonych (ang. Rough Set Theory) została zaproponowana przez Zdzisława Pawłaka na początku lat osiemdziesiątych [38]. Wprowadziła ona matematyczne podstawy do analizy dowolnych, dyskretnych danych zestawionych w tabeli (tablice informacyjne). Dane te traktowane jako przykłady dają podstawę do klasyfikacji zestawionych i nowych przypadków oraz wyznaczenia niezbędnych zbiorów atrybutów opisujących obiekty (również przy niepełnej i niepewnej informacji). Problemy klasyfikacji są szczególnie istotne w analizie i rozpoznawaniu obrazów, syntezie reguł decyzyjnych, ekstrakcji wzorców informacji z banków danych, aproksymacji informacji, redukcji informacji nadmiarowej itp.. Teoria zbiorów przybliżonych rozwija się dynamicznie znajdując zastosowania w coraz to nowych dziedzinach, że wspomnę kilka bardzo różnych dziedzin: medycyna, finanse i marketing, sterowanie produkcją, analiza obrazów i sygnałów, chemia czy socjologia (publikacje wyników można znaleźć między innymi w [45], [46]). Zastosowaniom tym towarzyszą systemy i moduły programowe. Szczegółowe opracowanie zarówno podstaw teoretycznych jak i pełną listę zastosowań teorii zbiorów przybliżonych można znaleźć w [29].

12.1 Podstawowe pojęcia

System informacyjny jest parą $I=(U, A)$, gdzie:

U – jest niepustym, skończonym zbiorem nazywanym *universum*; elementami zbioru są obiekty,

A – jest niepustym, skończonym zbiorem atrybutów.

Każdy atrybut $a \in A$ jest odwzorowaniem: $a: U \rightarrow V_a$, gdzie zbiór V_a jest zbiorem wartości atrybutu a .

Każdy obiekt $x \in U$ jest dokładnie opisany przez *informację* $Inf_A(x)$, gdzie $Inf_A(x) = \{(a, a(x)) : a \in A \text{ dla } x \in U\}$. Zbiór $\{Inf_A(x) : x \in U\}$ jest oznaczany przez INF_A i nazywany *zbiorem informacji* o x .

Każdy system informacyjny o postaci $I = (U, A \cup \{d\})$, nazywamy *tablicą decyzyjną*, atrybut $d \notin A$ jest nazywany *decyzją*, a atrybuty ze zbioru A *warunkami*.

Jak wynika z definicji systemu informacyjnego obiekty ze zbioru U są scharakteryzowane zbiorem atrybutów A . Konsekwentnie więc obiekty opisane takim samym wektorem informacyjnym są identyczne. Formalnie zostało to wyrażone poprzez relację *nierozróżnialności*.

Dla każdego niepustego podzbioru $B \subseteq A$, definiujemy relację *nierozróżnialności* w następujący sposób:

$$IND(B) = \{(x, y) \in U \times U : Inf_B(x) = Inf_B(y)\}.$$

Obiekty x, y będące w relacji $IND(B)$ są *nierozróżnialne* przez atrybuty ze zbioru B . Ponieważ relacja nierozróżnialności jest relacją równoważnościową to wprowadza ona podział obiektów na klasy równoważności: $[x]_B = \{y : x IND(B) y\}$ dla każdego $x \in U$. Relacja równoważności $IND(B)$ jest też dobrym narzędziem do aproksymacji podzbiorów obiektów z U . Dokładniej dla każdego $X \subseteq U$ definiujemy odpowiednio *dolne* i *górne przybliżenie* zbioru X następująco:

$$\underline{B}X = \{x \in U : [x]_B \subseteq X\}$$

$$\overline{B}X = \{x \in U : [x]_B \cap X \neq \emptyset\}$$

Parę $(\underline{B}X, \overline{B}X)$ nazywamy *zbiorem przybliżonym* zbioru X . Zbiór $BN_B(X) = \overline{B}X - \underline{B}X$ nazywamy regionem brzegowym – *brzegiem* zbioru X wyznaczonym przez zbiór B .

Zbiór $\underline{B}X$ interpretujemy jako zbiór wszystkich elementów zbioru U , które na podstawie informacji zawartych w podzbiorze B , mogą być na pewno sklasyfikowane jako elementy zbioru X . Natomiast zbiór $\overline{B}X$ zawiera wszystkie elementy zbioru U , które na

podstawie informacji zawartej w B mogą być prawdopodobnie sklasyfikowane jako elementy zbioru X . Elementy brzegu zbioru X z pewnością nie mogą być sklasyfikowane jako elementy zbioru X , ani zbioru $U-X$ (na podstawie B).

Przykład 12.1

W pracy znajduje się wiele przykładów tablic informacyjnych i tablic decyzyjnych. Pokażmy na przykładzie tablicy decyzyjnej (Tabela 9) podział obiektów na klasy równoważności dla podzbiorów atrybutów $B1=\{a\}$, $B2=\{b\}$ oraz $B3=\{a, b\}$.

$$\begin{aligned} IND(\{a\}) &= \{\{s_1, s_2\}, \{s_3, s_4, s_5\}\} \\ IND(\{b\}) &= \{\{s_1, s_3\}, \{s_2, s_4\}, \{s_5\}\} \\ IND(\{a, b\}) &= \{\{s_1\}, \{s_2\}, \{s_3\}, \{s_4\}, \{s_5\}\} \end{aligned}$$

Dolne i górne przybliżenie podzbioru $X=\{s_1, s_2, s_3\}$ dla wszystkich podzbiorów atrybutów będzie odpowiednio następujące:

$$\begin{aligned} \underline{B1X} &= \{s_1, s_2\}, \quad \overline{B1X} = \{s_1, s_2, s_3, s_4, s_5\} \\ \underline{B2X} &= \{s_1, s_3\}, \quad \overline{B2X} = \{s_1, s_2, s_3, s_4\} \\ \underline{B3X} &= \{s_1, s_2, s_3\}, \quad \overline{B3X} = \{s_1, s_2, s_3\} \end{aligned}$$

Otrzymane wyniki można interpretować w ten sposób, że dokładna klasyfikacja obiektów ze zbioru X jest możliwa, jeśli weźmiemy pod uwagę wszystkie atrybuty tzn. zbiór $B3$ ponieważ $BN_{B3}(X)=\emptyset$.

Uwaga 12 Zauważmy, że otrzymany wynik jest zgodny z wyznaczonym reduktom w przykładzie 3.1.6. Wyznaczając minimalne reguły decyzyjne wzięliśmy pod uwagę kolejno zbiory X równe $X1=\{s_1\}$, $X2=\{s_2\}$ i $X3=\{s_3\}$ i odpowiednio dla nich zbiory obiektów $U-X2-X3$, $U-X1-X3$ oraz $U-X1-X2$ (szczegóły sposobu wyznaczania minimalnych reguł decyzyjnych przedstawiono w punkcie 12.4).

12.2 Redukty

Redukty są bardzo ważnym pojęciem w teorii zbiorów przybliżonych. Są one najbardziej istotną częścią wiedzy – zbiorem tych atrybutów, które zachowują podział obiektów na klasy równoważności w relacji nierozróżnialności. Głównym zastosowaniem reduktów jest redukcja nadmiarowej informacji (usunięcie pozostałych atrybutów nie wpływa na klasyfikację obiektów) i ekstrakcja reguł decyzyjnych. Dokładnie; minimalny

podzbiór $B \subseteq A$ taki, że $IND(A) = IND(B)$ jest nazywany *reduktem* zbioru A . Zbiór wszystkich reduktów w A oznaczamy przez $RED(A)$.

Jądrem zbioru atrybutów A ($CORE(A)$) nazywamy część wspólną reduktów zbioru A tzn.: $CORE(A) = \bigcap RED(A)$.

Niech $I = (U, A \cup \{d\})$ będzie tablicą decyzyjną, założmy ponadto że zbiór wartości atrybutu decyzyjnego wynosi $V_d = \{v_d^1, \dots, v_d^{r(d)}\}$, gdzie $r(d)$ jest liczbą różnych wartości atrybutu decyzyjnego tzn. $r(d) = |\{i: d(x) = i, x \in U\}|$. Uogólniona decyzja w zbiorze A jest funkcją $\delta_A: U \rightarrow P(V_d)$ zdefiniowaną w następujący sposób:

$$\delta_A(x) = \{i: \exists x' \in U (x' \text{ } IND(A) \text{ } x) \wedge d(x') = i\} = d([x]_A)$$

Tablicę decyzyjną I nazywamy *spójną*, gdy $card(\delta_A(x)) = 1$ dla każdego $x \in U$. W przeciwnym razie (jeśli $card(\delta_A(x)) > 1$) tablica decyzyjna jest *niespójna*.

Zbiór atrybutów $B \subseteq A$ nazywamy *relatywnym reduktem* tablicy decyzyjnej I wtedy i tylko wtedy gdy:

1. dla każdego $x \in U$ $\delta_B(x) = \delta_A(x)$,
2. żaden podzbiór zbioru B nie spełnia warunku 1.

tzn. B jest minimalnym (ze względu na relację zawierania \subseteq) podzbiorem atrybutów mającym następującą własność $\forall x \in U \delta_B(x) = \delta_A(x)$.

Metodami wyznaczenia reduktów zajmowało się wielu autorów [41], [61]. Jeden ze sposobów oparty o wnioskowanie boolowskie, zostanie przedstawiony poniżej (punkt 12.3). Szczególne znaczenie (ze względu na zastosowania) mają dwa problemy: wyznaczenie minimalnych reduktów (ze względu na liczbę użytych atrybutów) i wyznaczenie wszystkich reduktów. Wysoka złożoność obliczeniowa obu problemów (wyznaczenie minimalnych reduktów jest NP – trudne) jest powodem szukania algorytmów heurystycznych. W jednym z nich zastosowano algorytmy genetyczne [69].

12.3 Wnioskowanie boolowskie, funkcja rozróżnialności

Wnioskowanie boolowskie [9], [65], [44], [41] jest bardzo często używane przy rozwiązywaniu wielu problemów teorii zbiorów przybliżonych. Metoda ta polega na zakodowaniu problemu optymalizacji π przez odpowiadającą funkcję boolowską f_π w taki sposób, że każdy pierwszy implikant tej funkcji jest rozwiązaniem problemu π . Metoda ta jest intensywnie wykorzystywana w pracy. Zilustrujemy tę metodę na przykładzie problemu wyznaczania reduktów z tablicy decyzyjnej.

Niech będzie dana tablica decyzyjna $I=(U, A \cup \{d\})$, gdzie $U=\{x_1, \dots, x_n\}$ i $A=\{a_1, \dots, a_k\}$. *Matrycą rozróżnialności* tablicy decyzyjnej I nazywamy macierzę rzędu $(n \times n)$:

$$M(I) = [C_{i,j}]_{i,j=1}^n$$

taką, że $C_{i,j}$ jest zbiorem atrybutów rozróżniających x_i od x_j . Formalnie:

$$C_{i,j} = \begin{cases} \{a \in A : a(x_i) \neq a(x_j)\} & \text{gd } d(x_i) \neq d(x_j) \\ \emptyset & \text{w przeciwnym razie} \end{cases}$$

Funkcją rozróżnialności f_I będziemy nazywali funkcję boolowską k zmiennych boolowskich $a_1^* \dots a_k^*$ (odpowiadających atrybutom a_1, \dots, a_k) zdefiniowaną w następujący sposób:

$$f_I(a_1^* \dots a_k^*) = \bigwedge_{1 \leq i, j \leq n} \left(\bigvee_{a \in C_{i,j}} a^* \right)$$

Zbiór wszystkich implikatów pierwszych tej funkcji wyznacza zbiór wszystkich reduktów tablicy decyzyjnej I . Można zauważyć, że zbiór $B \subset A$ jest reduktem jeśli ma on niepuste przecięcie z każdym niepustym zbiorem $C_{i,j}$. Formalnie:

$$B \text{ jest reduktem w } I \Leftrightarrow \forall_{i,j} (C_{i,j} \neq \emptyset) \vee (B \cap C_{i,j} \neq \emptyset)$$

W pewnych zastosowaniach, zamiast reduktów lepiej jest używać ich przybliżeń (aprosymacji). Niech będzie dana liczba rzeczywista $\alpha \in [0, 1]$ – wartość parametru jest

dobierana dynamicznie podczas dostrajania systemu i adekwatnie do zastosowania. Zbiór atrybutów B jest nazywany α -reduktem wtedy i tylko wtedy gdy $B \subseteq A$ ma niepuste przecięcie z co najmniej $\alpha \cdot 100\%$ niepustych zbiorów $C_{i,j}$, to znaczy:

$$B \text{ jest } \alpha\text{-reduktem w } I \Leftrightarrow \frac{|\{C_{i,j} : B \cap C_{i,j} \neq \emptyset\}|}{|\{C_{i,j} : C_{i,j} \neq \emptyset\}|} \geq \alpha$$

Można pokazać, że dla danego α problem wyznaczenia najkrótszego (minimalnego) α -reduktu i wszystkich α -reduktów jest obliczeniowo NP-trudny.

12.4 Minimalne reguły decyzyjne

Niech będzie dana tablica decyzyjna $I=(U, A \cup \{d\})$ i zbiór $V=\cup\{V_a: a \in A\} \cup V_d$. *Formuły atomowe* dla $B \subseteq A \cup \{d\}$ i V to wyrażenia typu $a=v$, zwane *deskryptorami* w B i V , gdzie $a \in B$ i $v \in V_a$. Zbiór $F(B, V)$ definiujemy jako najmniejszy zbiór zawierający wszystkie formuły atomowe dla zbioru B i V , domknięty ze względu na operatory \vee (dyzjunkcję), \wedge (koniunkcję) i \neg (negację).

Znaczenie formuł atomowych definiujemy rekurencyjnie. Niech $\varphi \in F(B, V)$, $|\varphi|_I$ będzie znaczeniem φ w tablicy decyzyjnej I , tzn. $|\varphi|_I$ jest zbiorem wszystkich obiektów z U o właściwości φ . Wtedy:

1. jeśli φ jest postaci $a=v$, to $|\varphi|_I = \{x \in U: a(x)=v\}$
2. $|\varphi \wedge \varphi'|_I = |\varphi|_I \cap |\varphi'|_I$; $|\varphi \vee \varphi'|_I = |\varphi|_I \cup |\varphi'|_I$; $|\neg \varphi|_I = U - |\varphi|_I$

Zbiór $F(B, V)$ jest nazywany zbiorem *formuł warunkowych* systemu I i jest oznaczany przez $C(B, V)$.

Regułą decyzyjną systemu I jest wyrażenie postaci $\varphi \Rightarrow d=v$ gdzie $\varphi \in C(B, V)$, $v \in V_d$ i $|\varphi|_I \neq \emptyset$. Reguła decyzyjna jest *prawdziwa* w I wtedy i tylko wtedy, gdy $|\varphi|_I \subseteq |d=v|_I$.

Generowanie reguły decyzyjnych na podstawie tablic decyzyjnych jest podstawą funkcjonowania wielu systemów wspomaganie decyzji. Najważniejszym problemem związanym z generowaniem reguły decyzyjnych jest wyznaczenie atrybutów użytych w

warunku. W teorii zbiorów przybliżonych atrybuty warunkowe są wyznaczane na podstawie reduktów. Ponieważ jest wiele typów reduktów, reguły decyzyjnych można wyznaczyć na wiele sposobów. W pracy są wykorzystywane *minimalne reguły decyzyjne*. Charakteryzują się one tym, że bardzo dobrze klasyfikują nowe obiekty (nie wymienione w tablicy decyzyjnej). Sposób wyznaczenia minimalnych reguł decyzyjnych dla systemu I i decyzji v tzn. $d(u_i)=v$ dla $u_i \in U$ i $v \in V_d$ można przedstawić w następujący sposób. Zakładamy, że tablica decyzyjna jest spójna (jeśli tablica decyzyjna jest niespójna wprowadzamy uogólnioną decyzję). Dla każdego obiektu u_i takiego, że $d(u_i)=v$:

- Tworzymy zmodyfikowaną tablicę rozróżnialności M taką, dla której:
 - jeśli $i \neq l$ to $C^*_{i,j} = \emptyset$
 - jeśli $C_{i,j} \neq \emptyset$ i $d(u_i) \neq v$ to $C^*_{i,j} = C_{i,j}$ w przeciwnym razie $C^*_{i,j} = \emptyset$
- Wyznaczamy funkcję rozróżnialności f_M i obliczamy implikanty pierwsze formuły, biorąc pod uwagę wszystkie niepuste elementy matrycy M .
- Każdy implikant pierwszy $a_{i1} \wedge \dots \wedge a_{ir}$ formuły f_M wyznacza regułę decyzyjną:
 - $a_{i1} = v_{i1} \wedge \dots \wedge a_{ir} = v_{ir} \Rightarrow d = v$,
 - gdzie $a_{ij}(u_i) = v_{ij}$, dla $j = 1, \dots, r$, $d(u_i) = v$.

12.5 Relacja tolerancji

W systemach informatycznych mamy do czynienia z rzeczywistymi wartościami atrybutów. Często w takich przypadkach relacja nierozróżnialności wprowadza zbyt wiele małych klas równoważności. Dlatego wprowadza się uogólnienie, osłabiające założenia na relację rozróżnialności przyjmując, że relacja rozróżnialności jest tolerancją.

Relacja $\tau \subseteq U \times U$ jest *relacją tolerancji* na zbiorze U wtedy i tylko wtedy gdy:

- $\forall x \in U \langle x, x \rangle \in \tau$ (zwrotność)
- $\forall x, y \in U (\langle x, y \rangle \in \tau \Rightarrow \langle y, x \rangle \in \tau)$ (symetryczność)

Używając relacji tolerancji można rozszerzyć pojęcie *nierozróżnialności* obiektów do pojęcia *podobieństwa*.

Niech będzie dana tablica decyzyjna $I = (U, A \cup \{d\})$, gdzie $A = \{a_1, \dots, a_k\}$. Najpierw zdefiniujemy relację tolerancji w $INF_A \times INF_A$, gdzie $INF_A = \prod_{a \in A} V_a$, a V_a jest zbiorem wartości

atrybutu $a \in A$. Następnie użyjemy relacji τ_A w INF_A dla zdefiniowania relacji tolerancji w $U \times U$.

$$\forall_{x, y \in U} \{ \langle x, y \rangle \in \tau \Leftrightarrow \langle Inf_A(x), Inf_A(y) \rangle \in \tau_A \}$$

Dla każdego $x \in U$ relacja tolerancji τ definiuje klasy tolerancji:

$$[x]_\tau = \{ y \in U : \langle x, y \rangle \in \tau \}$$

Jeśli $\langle x, y \rangle \in \tau$ to mówimy, że relacja tolerancji *identyfikuje* obiekty x, y w przeciwnym razie *odróżnia* je.

Można też zdefiniować dolne i górne przybliżenie podzbioru obiektów $X \subseteq U$ ze względu na relację tolerancji. Definiujemy te pojęcia odpowiednio w następujący sposób:

$$\underline{\tau}(X) = \{ x \in U : [x]_\tau \subseteq X \} \qquad \overline{\tau}(X) = \{ x \in U : [x]_\tau \cap X \neq \emptyset \}$$

Więcej na temat relacji tolerancji i jej zastosowań można znaleźć np. w [29], [45], [46], [47]. W pracy wprowadzono pojęcie podobieństwa stanów lokalnych, które jest relacją tolerancji (patrz 3.1.2). Można też zauważyć, nie wchodząc w szczegóły definicji, że koalicje tworzą klasy tolerancji. (Przyjaciel mojego przyjaciela może nie być moim przyjacielem.) Wiodącym zastosowaniem relacji tolerancji w zbiorach przybliżonych jest aproksymacja zbiorów, klasyfikacja często na podstawie relacji wydobytych, odkrytych z przykładowych danych [46].

BIBLIOGRAFIA

- [1] Aizerman, M. A. 1985, "New Problems in the General Choice Theory", *Social Choice Welfare* 2, pp. 235-282.
- [2] Angur, M., 1996, "A Hybrid Conjoint-Measurement and Bi-Criteria Model for a 2 Group Negotiation Problem", *Socio-Economic Planning Sciences* 30(3), pp. 195-206.
- [3] Arrow, K. J. 1963, "Social Choice and Individual Values" Wiley, New York.
- [4] Avouris, M.; Gasser, L., 1992, "Distributed Artificial Intelligence: Theory and Praxis", Boston, Mass.: Kluwer Academic.
- [5] Barber, K.S.; Liu, T.H.; Goel, A.; Martin C.E., 1999, "Conflict Representation and Classification in a Domain-Independent Conflict Management Framework", *Proceedings of the third annual conference on Autonomous Agents*, May 1-5, Seattle WA USA, pp. 346-347.
- [6] Beringer, B.; De Backer, B., 1998, "Combinatorial problem solving in Constraint Programming with cooperating Solvers", *Logic Programming: Formal Methods and Practical Applications*, C. Beirle and L. Palmer editors, North Holland.
- [7] Botelho, S.S.C., 1998, "A distributed scheme for task planning and negotiation in multi-robot systems", 13th ECAI. Edited by Henri Prade. Published by John Wiley & Sons, Ltd.
- [8] Brams, S.J.; Wittman, D., 1981, "Nonmyopic equilibria in 2x2 games" *Conflict Management and Peace Science* 6(1), pp. 39-62.
- [9] Brown, F. N., 1990, "Boolean Reasoning". Kluwer, Dordrecht.
- [10] Bui, T., 1994, "Software Architecture for Negotiator Support: Co-op and Negotiator", *Computer-Assisted Negotiation and Mediation Symposium*, Harvard Law School, Cambridge, MA.
- [11] Day W. H. E., 1988, "Consensus Methods as Tools for Data Analysis" in: *Classification and related methods for data analysis*, H. H. Back (ed.) North Holland 1988.
- [12] Deja, R., 1996, "Conflict Analysis", *Proceedings of the Fourth International Workshop on Rough Sets, Fuzzy Sets and Machine Discovery*, The University of Tokyo, 6-8 November, pp. 118-124.
- [13] Deja, R., 1996, "Conflict Model with Negotiation", *Bulletin of the Polish Academy of Sciences, Technical Sciences*, vol. 44, no. 4, pp. 475-498.

- [14] Deja, R., 1999, "Conflict Analysis", Proceedings of the 7th European Congress on Intelligent Techniques & Soft Computing, Aachen, Germany, September 13-16.
- [15] Deja, R., 2000, "Conflict Analysis", Rough Sets; New Developments. In: L. Polkowski (eds.), Studies in Fuzziness and Soft Computing, Physica-Verlag.
- [16] Ephrati E.; Rosenschein J. S., 1996, "Deriving Consensus in Multiagent Systems", Artificial Intelligence vol. 87, pp. 21-74.
- [17] Everitt, B. 1980. "Cluster Analysis". London, United Kingdom: Heinmann Educational Books, Second Edition.
- [18] Fang, L.; Hipel, K.W.; Kilgour, D.M., 1993, "Interactive Decision Making: the Graph Model for Conflict Resolution", Wiley, New York.
- [19] Fraser, N.M.; Hipel, K.W., 1984, "Conflict Analysis: Models and Resolutions", North-Holland, New York.
- [20] Fraser, N.M.; Hipel, K.W., 1983, "Dynamic modelling of the Cuba missile crisis", Journal of the Conflict Management and Peace Science 6 (2), pp. 1-18.
- [21] Grzymala-Busse, J., 1992, "LERS - a System for Learning from Examples Based on Rough Sets", In Słowiński R. [ed.] Intelligent Decision Support, Handbook of Applications and Advances of the Rough Sets Theory, Kluwer, pp. 3-18.
- [22] Hart, S.; Mas-Colell, A., 1997, "Cooperation: Game Theoretical Approach", Nalo ASI Series, Computer and System Sciences vol. 155.
- [23] Hipel, K.W.; Meiser, D.B., 1993, "Conflict analysis methodology for modeling coalition formation in multilateral negotiations", Information and Decision Technologies.
- [24] Howard, N., 1975, "Metagame analysis of business problems", INFOR 13, pp. 48-67.
- [25] Howard, N.; Shepanik, I., 1976, "Boolean algorithms used in metagame analysis", Univeristy of Ottawa. Canada.
- [26] Kersten, G.E.; Szpakowicz, S., 1994, "Negotiation in Distributed Artificial Intelligence: Drawing from Human Experiences", Proceedings of the 27th Hawaii International Conference on System Sciences. Volume IV, J.F. Nunamaker and R.H. Sprague, Jr. (eds.), Los Alamitos, CA: IEEE Computer Society Press, pp. 258-270.
- [27] Kersten, G.E.; Rubin, S.; Szpakowicz, S., 1994, "Medical Decision Making in Negoplan. Moving Towards Expert Systems Globally in the 21st Century", Proceedings of the Second World Congress on Expert Systems, J. Liebovitz (ed.) Cambridge, MA: Macnillan, pp. 1130-1137.

- [28] Kleinberg, J.; Papadimitriou Ch.; Raghavan P., 1998, "A microeconomic View of Data Mining", *Journal of Data Mining and Knowledge Discovery*, vol. 2, issue 4, pp. 311-324.
- [29] Komorowski, J.; Pawlak, Z.; Polkowski, L.; Skowron, A., 1999, Rough sets: A tutorial. in: S.K. Pal and A. Skowron (eds.), *Rough fuzzy hybridization: A new trend in decision making*, Springer-Verlag, Singapore, pp. 3-98.
- [30] Kumar, A.; Malik, K., 1996, "Optimising the Cost in Hierarchical Quorum Consensus", *Acta Informatica* 33, pp. 255-275.
- [31] Markov, I. M.; Vinograskaya, T. M.; Rubehinsky, A. A.; Sokolov, V. B. 1987, "The Theory of Choice and Decision Making", Mir, Moscow.
- [32] Nash, J.E., 1950, "Equilibrium Points in n-Person Games", *Proceedings National Academy of Science* 26, pp. 48-49.
- [33] Neigebauer, A., 1999, "Informatyczne Systemy Wspomagania Decyzji w Sytuacjach Konfliktowych", *Wojskowa Akademia Techniczna, Warszawa*.
- [34] Nęcki, Z., 1994, "Negotiations in business", Professional School of Business Edition. (The book in Polish), Krakow.
- [35] Von Neumann, J.; Morgenstern O., 1947, "Theory of Games and Economic Behaviour", Princeton.
- [36] Oswalt, I., 1993, "Current Applications, Trends and Organizations in U.S. Military Simulation and Gaming", *Simulation & Gaming*, June vol. 24, No. 2, pp. 53-189, Sage Publications, Inc.
- [37] Pawlak, Z., 1981, "Information Systems - Theoretical Foundations", (The book in Polish), PWN Warsaw.
- [38] Pawlak, Z. 1982, "Rough Sets", *Int. J. of Information and Computer Science* 11, pp. 344-356.
- [39] Pawlak, Z., 1984, "On Conflicts", *Int. J. of Man-Machine Studies*, 21, pp. 127-134.
- [40] Pawlak, Z., 1987, "O Konfliktach", Państwowe Wydawnictwo Naukowe, Warszawa.
- [41] Pawlak, Z., 1991, "Rough Sets - Theoretical Aspects of Reasoning about Data", Kluwer Academic Publishers, Dordrecht.
- [42] Pawlak, Z., 1993, "Anatomy of Conflicts", *Bull. EATCS*, 50, pp. 234-246.
- [43] Pawlak, Z., 1998, "An Inquiry into Anatomy of Conflicts", *Journal of Information Sciences* 109 pp. 65-78.

- [44] Pawlak, Z.; Skowron, A., 1993, "A Rough Set Approach to Decision Rules Generation", Institute of Computer Science Reports, 23/93, Warsaw University of Technology.
- [45] Polkowski, L.; Skowron, A., (Eds.) 1998, "Rough Sets in Knowledge Discovery 1: Methodology and Applications", Physica-Verlag, Heidelberg.
- [46] Polkowski, L.; Skowron, A., (Eds.) 1998, "Rough Sets in Knowledge Discovery 2: Applications, Case Studies and Software Systems", Physica-Verlag, Heidelberg.
- [47] Polkowski, L.; Skowron, A.; Żytkow, J., 1995, "Tolerance Based Rough Sets", Lin and Wildberger Eds. "Soft Computing: Rough Sets, Fuzzy Logic, Neural Networks, Uncertainty Management, Knowledge Discovery". Simulation Councils, Inc., San Diego, CA. pp. 55-58.
- [48] Prekopa, A., 1995, "Stochastic Programming", Kluwer Academic Publisher, Dordrecht/Boston/London.
- [49] Przemieniecki, J.S., 1994, "Mathematical Methods in Defence Analysis", second edition, AIAA, Washington DC.
- [50] Puget, J-F., 1998, "Constraint Programming: A great AI Success", 13th ECAI 98. Edited by Henri Prade. Published by John Wiley & Sons, Ltd.
- [51] Rosenheim, J.S.; Zlotkin, G., 1994, "Designing Conventions for Automated Negotiation", AI Magazine 15(3) pp. 29-46. American Association for Artificial Intelligence.
- [52] Rosenheim, J.S.; Zlotkin, G., 1994, "Rules of Encounter: Designing Conventions for Automated Negotiations among Computers", The MIT Press, Cambridge.
- [53] Sandholm, T., 1996, "Negotiation among Self-Interested Computationally Limited Agents", Ph.D. Dissertation. University of Massachusetts at Amherst, Department of Computer Science. 297 pages.
- [54] Sandholm, T. 1992, "Automatic Cooperation of Area-Distributed Dispatch Centers in Vehicle Routing", International Conference on Artificial Intelligence Applications in Transportation Engineering, San Buenaventura, California, pp. 449-467.
- [55] Sandholm, T.; Lesser, V., 1995, "Equilibrium Analysis of the Possibilities of Unenforced Exchange in Multiagent Systems", Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95), Montreal, Canada, pp. 694-701.
- [56] Sandholm, T.; Lesser, V., 1995, "Issues in Automated Negotiation and Electronic Commerce: Extending the Contract Net Framework",

- Proceedings of the International Conference on Multiagent Systems pp. 328-335. American Association for Artificial Intelligence.
- [57] Sandholm, T.; Lesser, V., 1997, "Coalitions among Computationally Bounded Agents", *Artificial Intelligence* 94(1), pp. 99-137, Special issue on Economic Principles of Multiagent Systems.
 - [58] Schehory, O.; Kraus, S., 1996, "A Kernel-oriented model for Coalition-formation in General Environments: Implementation and Results", *Proceedings of the National Conference on Artificial Intelligence, (AAAI-96)*, Portland.
 - [59] Selman, B.; Kautz H.; McAllester D., 1997, "Ten Challenges in Propositional Reasoning and Search", *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI- 97)*, Nagoya, Aichi, Japan.
 - [60] Shubik, M., 1983, "Mathematics of Conflicts", North Holland.
 - [61] Skowron, A., Rauszer, C., 1991, "The Discernibility Matrix and Functions in Information Systems", *Institute of Computer Science Reports, 1/91*, Warsaw University of Technology, and *Fundamenta Informaticae*.
 - [62] Sycara, K., 1996, "Coordination of Multiple Intelligent Softwareagents". *International Journal of Cooperative Information Systems* 5(2-3) pp. 181-211.
 - [63] Szczuka, M., 1999, "Metody symboliczne i sieci neuronowe w konstrukcji klasyfikatorów". *Praca doktorska, promotor A. Skowron, Uniwersytet Warszawski*.
 - [64] Tohme, F.; Sandholm, T., 1997, "Coalition Formation Processes with Belief Revision among Bounded Rational Self-Interested Agents", *Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97), Workshop on Social Interaction and Communityware*, Nagoya, Japan, August 25.
 - [65] Wegener, I., 1987, "The Complexity of the Boolean Function", Wiley, Stuttgart.
 - [66] Wellman, M., 1995, "A Computational Market Model for Distributed Configuration Design", *AI EDAM* 9 pp. 125-133, Cambridge University Press.
 - [67] Wiederhold, G., 1992, "Mediators in the Architecture of Future Information Systems", *IEEE Computer* 25(3) pp. 38-49.
 - [68] Wiweger, A., 1986 "On the Notation of a Conflict", *Bulletin of the Polish Academy of Sciences: Mathematics* 34, 5-6, pp. 381-391.
 - [69] Wróblewski, J., 1995, "Finding Minimal Reducts Using Genetic Algorithms", *Proceedings of the International Workshop on Rough*

Sets Soft Computing at Second Annual Joint Conference on Information Sciences (JCIS'95) P.P. Wang (Ed.) Wrightsville Beach, North Carolina, 28 September – 1 October, pp. 472-492.

- [70] Wu, S-H; Soo, V-W, 1999, "Game Theoretic Reasoning in Multi-agent Coordination by Negotiation with a Trusted Third Party", Proceedings of the third annual conference on Autonomous Agents, May 1-5, Seattle WA USA, pp. 56-61.
- [71] Ziarko, W. (Ed.), 1994, "Rough Sets, Fuzzy Sets and Knowledge Discovery (RSKD'93)". Workshops in Computing, Springer-Verlag & British Computer Society, London, Berlin.
- [72] Zlotkin, G.; Rosenchein, J., 1993, "The Extend of Cooperation in State-oriented Domains: Negotiations among Tidy Agents", Computers and Artificial Intelligence, 12(2) pp. 105-122.
- [73] Zlotkin, G.; Rosenchein, J., 1993, "Negotiation with Incomplete Information about Worth: Strict versus Tolerant Mechanism", Proceedings of the First International Conference on Intelligent and Cooperative Information Systems, pp. 175-184, Rotterdam, The Netherlands.
- [74] Żakowski, W., 1991, "On Conflicts and Rough Sets", Bulletin of the Polish Academy of Science, Technical Science, 39, 3/1991.
- [75] Żakowski, W., 1991, "Conflicts, Configurations, Situations and Rough Sets", Bulletin of the Polish Academy of Science, Technical Science, 4/1991.

INDEX

ρ - funkcja odległości	7
$\phi_a(\text{ag}, \text{ag}')$ funkcja zleżności między agentami	6
ϵ_{ag} - parametr bliskości	13
μ_{ag} - poziom akceptacji	14
$\tau_{\text{ag}, \epsilon}$ - relacja tolerancji	11
(pokojowa) przestrzeń rozwiązań	36
a - mapa $U \rightarrow V_a$	5
A - zbiór atrybutów	5
Ag - zbiór agentów	9
$Cg_{\text{Ag}}(S)$ - natężenie obiektywnego konfliktu globalnego	22
$Cl_{\text{Ag}}(s)$ - natężenie konfliktu lokalnego	21
$Cp_{\text{Ag}}(S)$ - współczynnik natężenia subiektywnego konfliktu globalnego	22
$cr(S)$ - wskaźnik zagrożenia konfliktem	48
$d(s, s')$ - funkcja odległości	12
e_{ag} - funkcja celu	11
F - dowolna funkcja $R^m_+ \rightarrow R_+$	13
f_r - strategiczna przestrzeń ruchów	47
$f_{s(\text{ag})}$ - formuła opisująca stan lokalny s agenta ag	36
$\text{fst}(a_i)$ - opis strategii dla a_i	45
f_Z - opisuje strategiczną przestrzeń sytuacji	45
$g_{\text{ag}}(s)$ - funkcja zadowolenia agentów	35
I_{ag} - systemem informacyjny	9
$\text{INF}^*(\text{ag}_i)$ - zbiór wszystkich wektorów informacyjnych	16
INF_{Ag} - zbiór wektorów informacji	9
$\text{Inf}_{\text{Ag}}(s)$ - wektor informacyjny	9
koalicje	42
l_{ag} - zwycięski podział łupów dla agenta ag	41
M_{Ag} - system z ograniczeniami	20
N_{ag} - zbiór pokojowa przestrzeni ruchów	36
$p(S)$ - globalna funkcja preferencji	19
podział łupów	40
$q: S_{\text{Ag}} \rightarrow R_{[0, 1]}$ - funkcja jakości	18
R^- - relacja konfliktu	6
R^+ - relacja zgody	6
R^0 - relacja neutralności	6
r_{ag} - przestrzeń rozwiązań	37
S - system informacyjny	5
S_{Ag} - sytuacja w Ag	16
$\text{Score}_{\text{Ag}}(t)$ - zbiór sytuacji dobrych	18
$\text{Spref}_{\text{Ag}}(h)$ - zbiór preferowanych sytuacji	20
strategie	43
sytuacja trwała	35
t'_{ag} - formuła opisująca zbiór celów ag powiększony o stany z bliskiego sąsiedztwa	28
T_{ag} - zbiór celów agenta ag	13
$tb(S)$ - współczynnik trwałości	35
U - universe, zbiór obiektów	5
U_{ag} - zbiór stanów lokalnych agenta ag	9
V_a - zbiór wartości atrybutu a	5

v_{ag} – formuła przestrzeń ruchów	36
$val(f)$ - wartościowanie formuł boolowskich $val(f(a_v \wedge a_v))=0$	16
w_{ag} - moc agentów	39
$Z(S)$ - zbiór sytuacji ze strategicznej przestrzeni S	45
<i>zwycięzca</i>	35
χ - relacja przejścia	34

