

где C, D, T – интервальные оценки затрат, дохода и продолжительности программы в целом; Z_{ni} – булева переменная $Z_{ni} = 1$, если i -й вариант проекта для реализации n -й задачи включен в состав программы и $Z_{ni} = 0$ – в противном случае; γ_{ni} – булева переменная $\gamma_{ni} = 1$, аналогична Z_{ni} .

При этом, для реализации n -й задачи должен быть выбран один проект

$$\sum_{i=1}^{i^n} z_{ni} = 1; \quad i = \overline{1, i^n}, \quad (2)$$

а значения частных критериев должны удовлетворять следующим неравенствам

$$\begin{aligned} \sum_{n=1}^{n'} \sum_{i=1}^{i^n} c_{ni} z_{ni} &\leq C^{\text{задан}}, \\ \sum_{n=1}^{n'} \sum_{i=1}^{i^n} d_{ni} z_{ni} &\geq D^{\text{задан}}, \\ \sum_{n=1}^{n'} \sum_{i=1}^{i^n} \tau_{ni} z_{ni} \gamma_{ni} &\leq T^{\text{задан}}, \end{aligned} \quad (3)$$

где $C^{\text{задан}}, D^{\text{задан}}, T^{\text{задан}}$ – заданные значения частных критериев – затраты, доход и продолжительность программы в целом.

Для решения задач (1)-(3) разработан метод декомпозиции программы на проекты.

Анализ задач программы и формирование для каждой из них альтернативных проектов в зависимости от предметной области и содержания задач. (По принципу формирования портфеля проектов). 2. Анализ технологической зависимости проектов (построение сетевого графика проектов программы и интеграция его с жизненным циклом программы). 3. Проверка достижимости критериев (1). 4. Если не найдены компромиссные значения показателей программы C, D, T то повторить пункты 1–3. 5. Если найдены компромиссные значения показателей программы C, D, T – задача решена. Утверждается архитектура программы.

Таким образом, предложенная модель и метод позволяет формализовать процесс декомпозиции программы на проекты выбрать рациональную архитектуру программы, которая удовлетворяет заданным критериям и повысит эффективность управления интеграцией программы. Предложенная модель (1)-(3) относится к задачам линейного программирования с булевыми переменными.

Каргін А.О. (УкрДУЗТ, м. Харків),
Галіч Г.Б. (ДонНУ, м. Вінниця)

МОДЕЛЮВАННЯ ДЕЯКИХ ФУНКЦІЙ РОЗУМНОЇ МАШИНИ, СТВОРЕННІ ЗА КОНЦЕПЦІЄЮ ГРАНУЛЯРНИХ ОБЧИСЛЕНЬ

Розумні машини, що застосовуються у вигляді побутових пристроїв автоматизують функції, які раніше виконувала людина. Такі пристрої визначають та оминають перешкоди які зустрічають на своєму шляху. В доповіді розглядається новий підхід до створення чистючого пристрою типу “Cleaner”. Пристрій при виконанні завдання рухається по замкнутому простору (кімнаті). Під час руху ідентифікує ділянку забруднення її форму та тип забруднення. Відповідно до ідентифікованого типу забруднення він обирає оптимальний засіб чистки, що дозволить збільшити його ефективність. Після того як пристрій визначив тип забруднення та обрав оптимальну технологію чистки, вмикаються чистючі пристрої робота і він рухається по ділянці забруднення по розрахованій траєкторії.

Найбільші труднощі викликає завдання виявлення класу (прототипу форми), до якого відноситься форма ділянки, так, як це завдання потребує свого вирішення на підставі даних, що отримує машина при обході пята забруднення. Адже, ділянка забруднення буде не завжди правильної геометричної форми тому основна задача полягає у ідентифікуванні форми пята та, відповідно до неї, вибору траєкторії обходу під час чистки.

Розробка пристрою поділяється на наступні задачі: виявлення місця забруднення; виявлення типу забруднення; вибір траєкторії проходження роботом пята забруднення. У докладі розглядається перша та частково третя задачі.

Ідентифікація події, що робот перетинає фрагмента кордону пята, моделювалось, як перетин полоси під різними кутами. Інформація про це збирається від матриці датчиків відображення. Фрагмент гранулярної структури, що відбиває знання про перетин полоси під прямим кутом наведено на рисунку 1.

Дослідження проведено для різних прототипів. Розглядалися *елементарні статичні* прототипи, побудовані тільки на гранулах нульового рівня. Наприклад,

$$4 = \{ 1^{+1}-10^{+1}, 11^{+1}-20^{+1}, 21^{+1}-30^{+1}, 31^{+1}-40^{+1}, 41^{+1}-50^{+1}, 51^{+1}-60^{+1}, 61^{+1}-70^{+1}, 71^{+1}-80^{+1} \}$$

Крім повних прототипів ситуації, побудованих на свій множені гранул нульового рівня, були перевірені різні варіанти прототипів, визначених на підмножинах цих гранул. Приведено декілька таких прототипів, визначених на найбільш інформативних елементах.

$$4' = \{ 41^{+1}-50^{+1}, 51^{+1}-60^{+1}, 61^{+1}-70^{+1}, 71^{-1}-80^{-1} \},$$

$$4'' = \{ 51^{+1}-60^{+1}, 61^{+1}-70^{+1}, 71^{-1}-80^{-1} \},$$

$$4''' = \{ 61^{+1}-70^{+1}, 71^{-1}-80^{-1} \}.$$

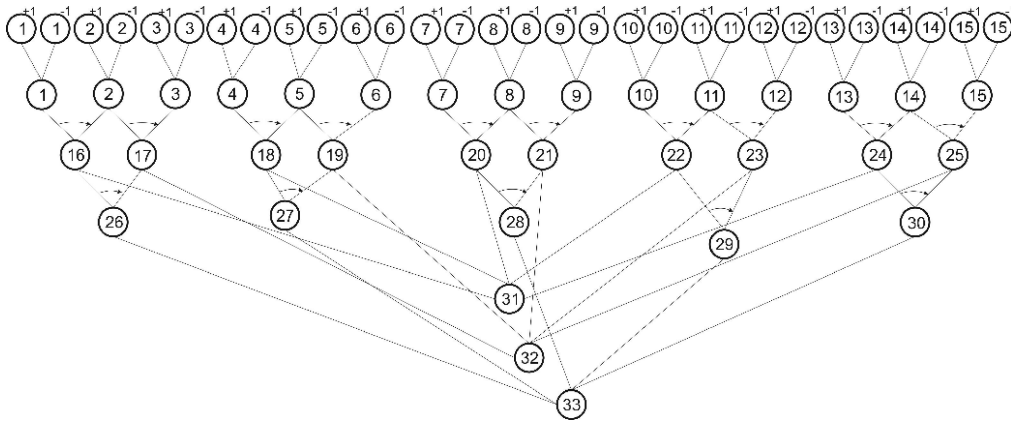


Рис. 1. Фрагмент гранулярної структури

Комп'ютерні експерименти полягали в наступному. Робот рухався зі сталою швидкістю і напрямом, які відповідають динамічним характеристикам прототипу № 4 і фіксувалися нечіткі характеристики гранул структури рис. 1. Далі використовуються різні варіації напрямку руху. Таким

чином, отримано сімейство залежностей нечітких характеристик від часу руху, що наведені на рис. 2.

З рисунків видно, що фактична ситуація, яка відповідає прототипу, добре локалізується у часі і просторі множини прототипів.

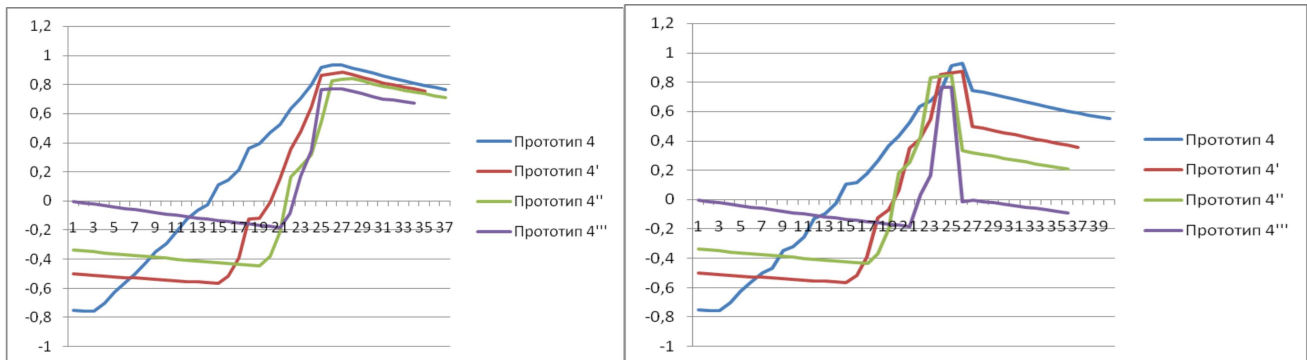


Рис. 2. Значення параметру впевненості нечіткої характеристики гранули № 33

Каргин А.А.

(Український державний університет залізничного транспорту, г. Харків),

Петренко Т.Г. (Донецький національний університет, г. Вінниця)

ПОДХОД К СОЗДАНИЮ УМНЫХ МАШИН, ОСНОВАННЫЙ НА МОДЕЛИ ГРАНУЛЯРНЫХ ВЫЧИСЛЕНИЙ

Одно из наиболее интенсивно развивающихся направлений, находящееся на стыке теории управления и искусственного интеллекта, нацелено на

создание умных машин [1]: начиная от умного дома и заканчивая умными автомобилями, пылесосами, уборщиками мусора, газонокосилками и другими бытовыми и машинами специального назначения. Ведущие фирмы мира вкладывают большие средства в проекты по созданию умных машин.

В ранних проектах интеллектуальных машин функции, которые выполняет искусственный интеллект, в основном, реализованы на моделях вычислительного интеллекта (нейро- и нечёткие регуляторы, простые и адаптивные), а также логики (продукционные системы, псевдофизические логики) и ситуационного управления (системы обычных и