Алгоритмы поиска с бикритериальной оптимизацией

Проект по курсу "Эвристические методы планирования"

Угадяров Л.А.
https://github.com/ugadiarov-la-phystech-edu/hs-project

МФТИ, группа М05-006а

15 апреля 2021 г.

Задачи многокритериальной оптимизации

Необходимо найти оптимальное решение, учитывая множество критериев.

Практические приложения:

- Прокладка телекомуникационных сетей: стоимость и вероятность отказа
- Планирование в робототехнике: длинна пути, потребление энергии
- Езда на велосипеде: длинна пути, безопасность велосипедиста
- Грузоперевозки: стоимость транспортировки, время в пути, экологические факторы
- Перевозки опасных грузов: длина пути, риск человеческих жерт при возможной аварии
- Пассажирские перевозки: стоимость проезда, время в пути, количество пересадок
- Планирование спутниковой фотосъёмки: удовлетворение запросов пользователей, приоритет запросов, минимизация износа оборудования

Математическая постановка бикритериальной задачи

Пусть $\mathbb{p} = (p_1, p_2)$ и $\mathbb{q} = (q_1, q_2)$ — пары вещественных чисел, тогда:

- $\mathbb{p} \prec \mathbb{q}$ ($\mathbb{p}$ доминирует $\mathbb{q}$), если $(p_1 < q_1) \wedge (p_2 \leq q_2)$ или $(p_1 = q_1) \wedge (p_2 < q_2)$
- $\mathbb{p} \preceq \mathbb{q}$ ($\mathbb{p}$ слабо доминирует $\mathbb{q}$), если $(p_1 \leq q_1) \wedge (p_2 \leq q_2)$

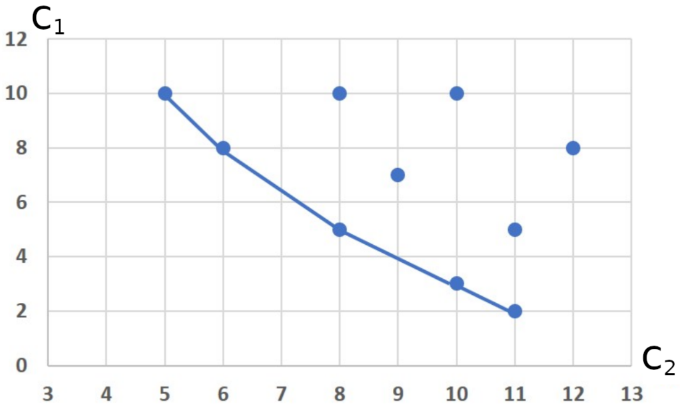
Бикритериальная задача поиска маршрутов с наименьшей стоимостью $\mathcal{U} = (S, E, \mathbb{h}, \mathbb{c}, s_{start}, s_{goal})$:

- S — конечное множество состояний
- $E \subseteq S \times S$ — множество рёбер
- $\mathbb{c} : E \to \mathbb{R}^+ \times \mathbb{R}^+$ — функция стоимости, $\mathbb{c}(e) = (c_1(e), c_2(e))$
- $\mathbb{h} : S \to \mathbb{R}^+ \times \mathbb{R}^+$ — эвристическая функция, $\mathbb{h}(s) = (h_1(s), h_2(s))$
- s_{start} — начальное состояние
- s_{goal} — целевое состояние

$\pi(s_1, s_n) = s_1, \ldots, s_n$ — маршрут из s_1 в s_n, где $\{s_i\} \subseteq S$ и $\{(s_i, s_{i+1})\} \subseteq E$.
$\mathbb{c}(\pi) = \sum_{i=1}^{n-1} \mathbb{c}(s_i, s_{i+1})$ — стоимость маршрута π.
Маршрут $\pi(s_1, s_n)$ доминирует маршрут $\pi^{'}(s_1, s_n) : \pi \prec \pi^{'} \Leftrightarrow \mathbb{c}(\pi) \prec \mathbb{c}(\pi^{'})$.
Маршрут $\pi(s_{start}, s_{goal})$ называется парето-оптимальным решением задачи $\mathcal{U} \Leftrightarrow \nexists \pi^{'}(s_{start}, s_{goal}) : \pi^{'} \prec \pi$.

Рассматривается поиск множества всех парето-оптимальных решений задачи $\mathcal{U}$ с уникальной стоимостью.
Рассматриваются монотонные эвристические функции: $\mathbb{h}(s_{goal}) = (0, 0)$ **и** $\forall (s, t) \in E \;\; \mathbb{h}(s) \preceq \mathbb{c}(s, t) + \mathbb{h}(t)$.

Пример парето-оптимального множества решений



$$(10, 5) \prec (10, 8),\ (10, 5) \not\prec (8, 6)$$

Общий подход к решению бикритериальной задачи

OPEN:

- *OPEN* содержит нераскрытые узлы — кортежи $x = (s, g, f)$, где g и f – векторы
- Для состояния s в *OPEN* одновременно могут находится несколько узлов (s, g_1, f_1) и (s, g_2, f_2)

Выбор узла из *OPEN*:

- Для раскрытия выбирается такой узел $(s, g, f) \in OPEN$, что $\nexists (s', g', f') \in OPEN : f' \prec f$
  Для бикритериальной задачи это условие эквивалентно извлечению узла с лексикографически минимальным значением f

Обработка узлов вида (s_{goal}, g, f):

- Поддерживается множество найденный решений: $SOL = \{\pi_i(s_{start}, s_{goal})\}$
- Если для найденного маршрута $\pi(s_{start}, s_{goal}) \Rightarrow \nexists \pi' \in SOL : \pi' \prec \pi$, то удаляем из *SOL* все маршруты $\tilde{\pi} : \pi \prec \tilde{\pi}$ и добавляем π в *SOL*

Раскрытие узла (s, g, f):

- Для каждого дочернего состояния $s' \in Succ(s)$ строится узел (s', g', f')
- Дочерний узел (s', g', f') добавляется в *OPEN* при одновременном выполнении двух условий:
  - $\nexists (s', \tilde{g}, \tilde{f}) \in OPEN : \tilde{f} \prec f'$
  - $\nexists \pi \in SOL : \mathbb{c}(\pi) \prec f'$
- Если (s', g', f') добавляется в *OPEN*, то из *OPEN* удаляются все узлы $(s', \tilde{g}, \tilde{f}) : f' \prec \tilde{f}$

Если *OPEN* пустой, то алгоритм завершает работу и возвращает *SOL*.

Алгоритмы

Алгоритм NAMOA*:

- Реализация общего подхода с незначительными оптимизациями за счёт поддержки множества всех раскрытых узлов $G_{cl}(s)$

Алгоритм NAMOA*dr

- Оптимизация операции добавления дочернего узла в $OPEN$ для случая монотонной эвристической функции и извлечения узлов из $OPEN$ в лексикографическом порядке по f:
  - Проверка $\nexists \pi \in SOL : \mathbb{c}(\pi) \prec (f_1^{'}, f_2^{'})$ заменяется на $\min_{\pi \in SOL} c_2(\pi) \geq f_2^{'}$
  - В некоторых случаях проверку $\nexists (s^{'}, \tilde{g}, \tilde{f}) \in OPEN : \tilde{f} \prec f^{'}$ можно можно заменить на $\min_{\tilde{f} \in OPEN} \tilde{f}_2 \geq f_2^{'}$

Алгоритм BOA*:

- Для случая монотонной эвристической функции и извлечения узлов из $OPEN$ в лексикографическом порядке по f доказаны ещё более сильные утверждения, которые позволяют все проверки доминирования совершать за константное время за счёт поддержки $g_2^{min}(s)$ — минимального значения g_2 для раскрытых узлов с состоянием s
- В узлах дополнительно хранятся ссылки на родительские узлы: $x = (s, g, f, parent(x))$

Mandow, L., and Pérez-de-la-Cruz, J. (2010). Multiobjective A* search with consistent heuristics
Machuca, E., and Mandow, L. (2012). Multiobjective heuristic search in road maps
Hernández Ulloa, C., Yeoh, W., Baier, J. A., Zhang, H., Suazo, L., & Koenig, S. (2020). A Simple and Fast Bi-Objective Search Algorithm

Bi-Objective A* (BOA*)

Сравнение качества работы алгоритмов на дорожных картах городов США 9th DIMACS Implementation Challenge - Shortest Paths. Критерии — длинна маршрута и время движения по маршруту:

- NAMOA*, NAMOA*dr, BOA* — авторские реализации (язык Си)
- sBOA* — модификация BOA* без применения оптимизаций, приводящих к константному времени проверки условий доминирования, авторская реализация (язык Си)
- Оригинальные реализации на Си для Bi-Objective Dijkstra (BDijkstra) и Bidirectional Bi-Objective Dijkstra (BBDijkstra) предоставлены авторами этих алгоритмов

New York City (NY)				
264,346 states, 730,100 edges, $\lvert sols \rvert$ = 199 on average				
	Solved	Average	Max	Min
NAMOA*	50/50	157.17	1,936.36	**0.02**
sBOA*	50/50	9.75	148.65	0.10
NAMOA*dr	50/50	0.65	4.99	0.11
BOA*	50/50	**0.32**	**1.95**	0.11
BBDijkstra	50/50	1.94	23.43	0.26
BDijkstra	50/50	2.55	21.16	0.17

Colorado (COL)				
435,666 states, 1,042,400 edges, $\lvert sols \rvert$ = 427 on average				
	Solved	Average	Max	Min
NAMOA*	48/50	476.26	3,551.32	**0.08**
sBOA*	50/50	38.88	1,141.78	0.17
NAMOA*dr	50/50	2.16	57.40	0.17
BOA*	50/50	**0.79**	**15.26**	0.17
BBDijkstra	50/50	4.79	83.07	0.41
BDijkstra	50/50	7.78	135.24	0.29

San Francisco Bay (BAY)				
321,270 states, 794,830 edges, $\lvert sols \rvert$ = 119 on average				
	Solved	Average	Max	Min
NAMOA*	50/50	58.87	1,474.76	**0.02**
sBOA*	50/50	3.38	120.57	0.12
NAMOA*dr	50/50	0.38	6.08	0.12
BOA*	50/50	**0.29**	**4.17**	0.12
BBDijkstra	50/50	0.87	9.61	0.28
BDijkstra	50/50	1.83	33.39	0.22

Florida (FL)				
1,070,376 states, 2,712,798 edges, $\lvert sols \rvert$ = 739 on average				
	Solved	Average	Max	Min
NAMOA*	43/50	812.48	3,298.90	1.42
sBOA*	46/50	349.64	1,238.25	**0.43**
NAMOA*dr	50/50	19.66	329.79	**0.43**
BOA*	50/50	**4.59**	**60.54**	**0.43**
BBDijkstra	50/50	91.36	1,772.48	1.11
BDijkstra	50/50	158.33	2,722.69	0.77

Algorithm : Bi-Objective A* (BOA*)

Input : A search problem $(S, E, \mathbf{c}, s_{start}, s_{goal})$ and a consistent heuristic function $\mathbf{h}$
Output: A cost-unique Pareto-optimal solution set

$sols \leftarrow \emptyset$
for each $s \in S$ **do**
  $g_2^{\min}(s) \leftarrow \infty$
$x \leftarrow$ new node with $s(x) = s_{start}$
$\mathbf{g}(x) \leftarrow (0, 0)$
$parent(x) \leftarrow$ null
$\mathbf{f}(x) \leftarrow (h_1(s_{start}), h_2(s_{start}))$
Initialize $Open$ and add x to it
while $Open \neq \emptyset$ **do**
  Remove a node x from $Open$ with the lexicographically smallest f-value of all nodes in $Open$
  if $g_2(x) \geq g_2^{\min}(s(x)) \vee f_2(x) \geq g_2^{\min}(s_{goal})$ **then**
    continue
  $g_2^{\min}(s(x)) \leftarrow g_2(x)$
  if $s(x) = s_{goal}$ **then**
    Add x to $sols$
    continue
  for each $t \in \mathtt{Succ}(s(x))$ **do**
    $y \leftarrow$ new node with $s(y) = t$
    $\mathbf{g}(y) \leftarrow \mathbf{g}(x) + \mathbf{c}(s(x), t)$
    $parent(y) \leftarrow x$
    $\mathbf{f}(y) \leftarrow \mathbf{g}(y) + \mathbf{h}(t)$
    if $g_2(y) \geq g_2^{\min}(t) \vee f_2(y) \geq g_2^{\min}(s_{goal})$ **then**
      continue
    Add y to $Open$
return $sols$

План работы

Репозиторий проекта: `https://github.com/ugadiarov-la-phystech-edu/hs-project`

1. Набор данных для задачи бикритериальной оптимизации — 9th DIMACS Implementation Challenge - Shortest Paths:
   - `http://www.diag.uniroma1.it/challenge9/download.shtml`

2. Формирование размеченной выборки для проверки корректности реализуемых алгоритмов поиска множества парето-оптимальных решений:
   - Использование алгоритма Дейкстры для поиска множества кротчайших путей по каждому критерию между парами вершин (пакет NetworkX: `https://networkx.org/`)
   - Построение множества парето-оптимальных решений по результатам работы алгоритма Дейкстры

3. Реализация BOA*:
   - Проверка корректности реализации на размеченной выборке
   - Эксперименты для оценки эффективности реализации: количество порождённых узлов, время работы
   - Оформление результатов

   Срок: 26.04.2021

4. Реализация NAMOA*:
   - Проверка корректности реализации на размеченной выборке
   - Эксперименты для оценки эффективности реализации: количество порождённых узлов, время работы
   - Оформление результатов

   Срок: 03.05.2021