ЗАДАЧА О КАНАДСКОМ ПУТЕШЕСТВЕННИКЕ НА ВНЕШНЕПЛАНАРНЫХ ГРАФАХ

Аннотация

В мире современной науки и информационных технологий, вопросы оптимального перемещения стали неотъемлемой частью многих областей. В этом контексте, исследование проблемы путешественника на внешнепланарных графах привлекает внимание как актуальная и сложная задача. Внешнепланарные графы представляют собой подмножество графов, в которых вершины и рёбра размещаются таким образом, что они не пересекаются в плоскости, и все вершины лежат на внешней грани графа. Проблема путешественника на таких графах затрагивает вопросы маршрутизации, оптимизации и навигации, имея широкие приложения в областях как математики и информатики, так и реальных практических сценариях, таких как планирование маршрутов в сетях связи и транспортных системах. В данной статье мы рассмотрим основные аспекты проблемы путешественника на внешнепланарных графах, анализируя существующие подходы к её решению и исследуя их применимость в различных контекстах.

Ключевые слова Внешнепланарный граф · Задача о канадском путешественнике

Введение

Мы работаем над взвешенным внешнепланарным графом $G = (V, E, w)$, $s, t \in V$.

Набор заблокированных рёбер : $E_k \subset E$, $|E_k| \leq k$.

Граф G/E_K не соединяет вершины от s до t

Рассматривая граф G с его внешнепланарной структурой, он допускает внешнюю грань, примыкающую ко всем вершинам. Мы выделяем две стороны внешней грани, к примеру:

от s до t по часовой стрелке: F_A, и от t до s: F_B.

$s, t \in F_A$, F_B и $F_A \cap F_B = \{s, t\}$, $F_A \cup F_B = \{V\}$

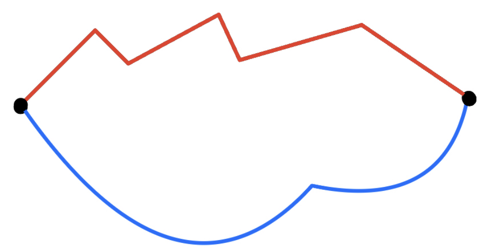

Рис. 1. red - F_A; blue - F_B.

Jmps и traversing chords.

Jmp является ребром, соединяющим две вершины из F_A или две вершины из F_B, которые не содержат s или t соотвественно.

Travesing chord это ребро, соединяющее вершину из F_A с вершиной из F_B. Travesing chords $\subseteq F_A \times F_B$.

Свойства внешнепланарного графа.

Пересекающая хорда, которая не содержит s или t, обозначим как $(x, y) \in (F_A, F_B) \cap E$ и пусть $(s, t)-$ разделителем

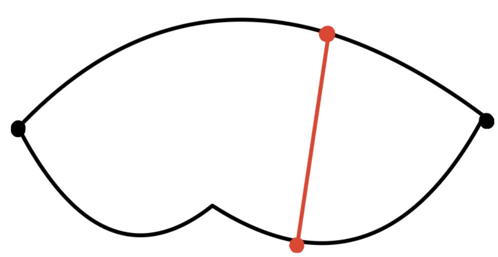

Рис. 2. red - (x, y).

вершины s и t разъединены в графе $G/\{x, y\}$.

Алгоритм

Текущий граф и кратчайший путь.

Учитывая перемещения на графе G, обозначим:

- E'_k the set of blockages discovered ($E'_k \supset E_k$);
- P_{min} кратчайший (s,t)-путь из $G/E_k(G_{curr})$ и w_{min} его веса (значение w_{min} увеличено);
- G_{curr} текущий граф: $G_{curr} = G/E'_k$;
- Для любой пары (u,v) вершин, $P(u \to v)$ является кратчайшим (u,v)-путем на графе G_{curr}. Пусть $w(u \to v)$ это вес;
- T_{trav}: подграф(неиндуцированный) графа G, содержащий ребра пройденные объектом.

Согласно нашей стратегии, T_{trav} является деревом.
По определению, $T_{trav} \cap E_k = \emptyset$.

Согласно нашей стратегии (описанной позже) ребро будет пройдено либо один раз, либо два раза(в обоих направлениях) во время каждого *"экспонециального шага*

.
$T_{trav} = T^1_{trav} \cup T^2_{trav}$ где T^1_{trav} пройдено единожды и T^2_{trav} пройдено дважды.
Обозначим через w_{trav} (μ, η) стоимость простого пути между $u \in T_{trav}$ и $v \in T_{trav}$ в дереве.

Algorithm 1 Основной Алгоритм

Input
G, s, t
Обозначим $w^*:= w_{min}$
while путешественник не доберется до t **do**
Шаг(G_{curr}, z, t, w^*) ▷ Где G_{curr} обновляется при обнаружении блокировки. w^* указывает на доступимую стоимость для перемещения (мы учитываем только пути, где стоимость находится между w^* and $2w^*$)).
end while

Каждый экспоненциальный шаг: идея состоит в том, чтобы рассмотреть пути из s в t со стоимостью между w^* и $2w^*$, где w^* это граница равная w_{min} в начале.

Algorithm 2 Шаг

Input

G, s, t, w^*

while путешественник не доберется до t ИЛИ существуют (s,t)-пути стоимостью $< 2w^*$ **do**

$P_{eff} \leftarrow P_{min}$

Пересекать P_{eff} до тех пор, пока не произойдет блокировка ▷ предположим, что узнали, что путешественник заблокирован на F_A

E1

$(v,w) \leftarrow$ maximal jmp $s.t. v \in T_{trav}, w \in P_{eff}/T_{trav}, (v,w)$ открыто и $w(s,v)_{trav} + w(v,w) + w(w,t) < 2w^*$ ▷ стоимость кратчайшего (s,t)-пути обязательно открытый и проходящий через (v,w)

if (v,w) существует **then**

Вернуться назад по T_{trav} в направлении v

$P_{eff} \leftarrow T_{trav}(s,v) \cup (v,w) \cup P(w \to t)$

Пройдите P_{eff} от v до t

if заблокировано **then**

Вернуться к E1

end if

end if

$u \leftarrow$ местонахождение путешественника ▷ $u \in T_{trav}$

Возвращайтесь к $T_{trav}(s,u)$ до тех пор, пока не встретим вершина $x \in T_{trav}(s,u)$ такая, что:

- любой путь $T_{trav}(s,x) \cup (x \to t)$ имеет стоимость $< 2w^*$ Сах A
- или $x \in F_B$ Сах B

if x существует И Сах A **then**

$y \leftarrow$ вершина s.t. (x, y) является первым ребром на пути $P(x \to t)$ ▷ (x,y) обязательно является travesty chord

$G' \leftarrow G$ без вершины x and the source side of separator $\{x,y\}$ ▷ G' - это внешнепланарный подграф G

Step(G', y, t, w^*) ▷ Идея состоит в том, чтобы "использовать ресурс"в вершине y. По индукции, мы знаем, что все пути от s до t со стоимостью $< 2w^*$ будут проходить через y.

end if

if x существует И Сах B **then**

$z \leftarrow$ successor of x on P_{eff} ▷ (x,z) является travesty chord

$G' \leftarrow G$ без исходной стороны сепаратора $\{x,z\}$

Шаг(G', z, t, w^*)

end if

if x не существует **then**

Стоп ▷ мы переходим к следующему шагу, и мы знаем, что $w_{min} \geq 2w^*$

end if

end while