Богдан Уладзіслаў

ФПМІ, 3 курс, 3 група

Практычны эксперымент па тэме артыкула

Peter Brucker, Rong Qu, Edmund Burke
Personnel scheduling: Models and complexity
Кадравае планаванне: мадэлі і складанасць

Пастаноўка задачы і апісанне алгарытма

Агулам, артыкул носіць класіфікацыйны і тэарэтычны характар. Для рэалізацыі эксперыментальнай часткі, разгледзім Лемму 1:

Спецыяльны выпадак прапанаванай матэматычнай мадэлі, у якой попыт кожнай працы - пастаянны, і кожны супрацоўнік даступны ў любы час, можа быць сфармуляваны як задача пошуку плыні мінімальнага кошту.

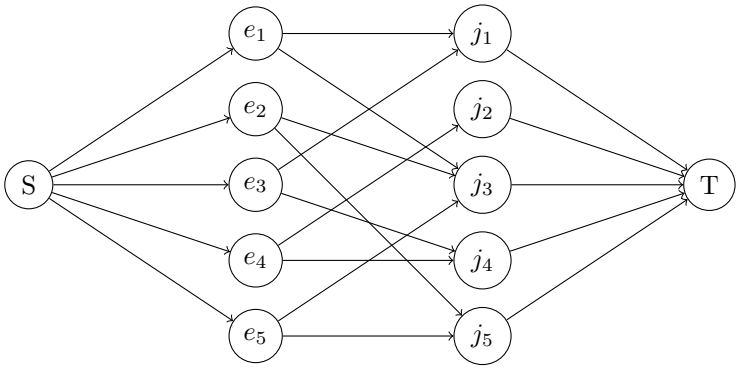
У артыкуле прыводзіцца канструктыўны доказ: сцвярджаецца, што рашэнне задачы прывядзецца да рашэння задачы пошуку плыні мінімальнага кошту, калі для кожнага супрацоўніка ўвесці вяршыню e, для кожнай працы вяршыню j, для кожнай пары (e, j) праведзеная дуга з коштам c_{ej} тады і толькі тады, калі $j \in Q_e$, дзе Q_e - мноства працаў, на якія кваліфікаваны супрацоўнік e.

Прывядзем апісанне нашай задачы. Маем часавы гарызонт (часавую стужку) $[0, T]$ падзелены на перыяды $[t, t+1[, t = 0, 1, ..., T-1$, за гэты час m працаў павінныя быць выкананыя. $D_j(t) = D_j, j = 1, ..., m$ - колькасць супрацоўнікаў, патрэбная, каб выканаць працу j. Маем мноства E з n супрацоўнікаў, з кожным супрацоўнікам асацыяванае мноства Q_e задачаў, на якія гэты супрацоўнік кваліфікаваны. Кожны супрацоўнік даступны ў любы час, то бок $w_e(t) = 1, \forall e \in Q_e, \forall t \in [0, T]$. Шаблон працы для кожнага супрацоўніка ўяўляе сабой вектар $\pi = \pi(j, t)$, дзе $\pi(j, t) = 1$ тады і толькі тады, калі ў перыяд часу $[t, t+1[$ супрацоўнік будзе выконваць працу j.

Мэта - мінімізаваць суму штрафаў $\sum c_{ej}$

Такім чынам, нам трэба скласці такі расклад, дзе кожны супрацоўнік робіць максімум адну працу (але на працягу ўсяго часу), бо пры нашых умовах мяняць працу аднаму супрацоўніку не дае ніякай перавагі, кожная праца выконваецца роўна D_j супрацоўнікамі, пры гэтым сумарны кошт працы - мінімальны.

Задача можа быць праілюстраваная графам наступнага выгляду:



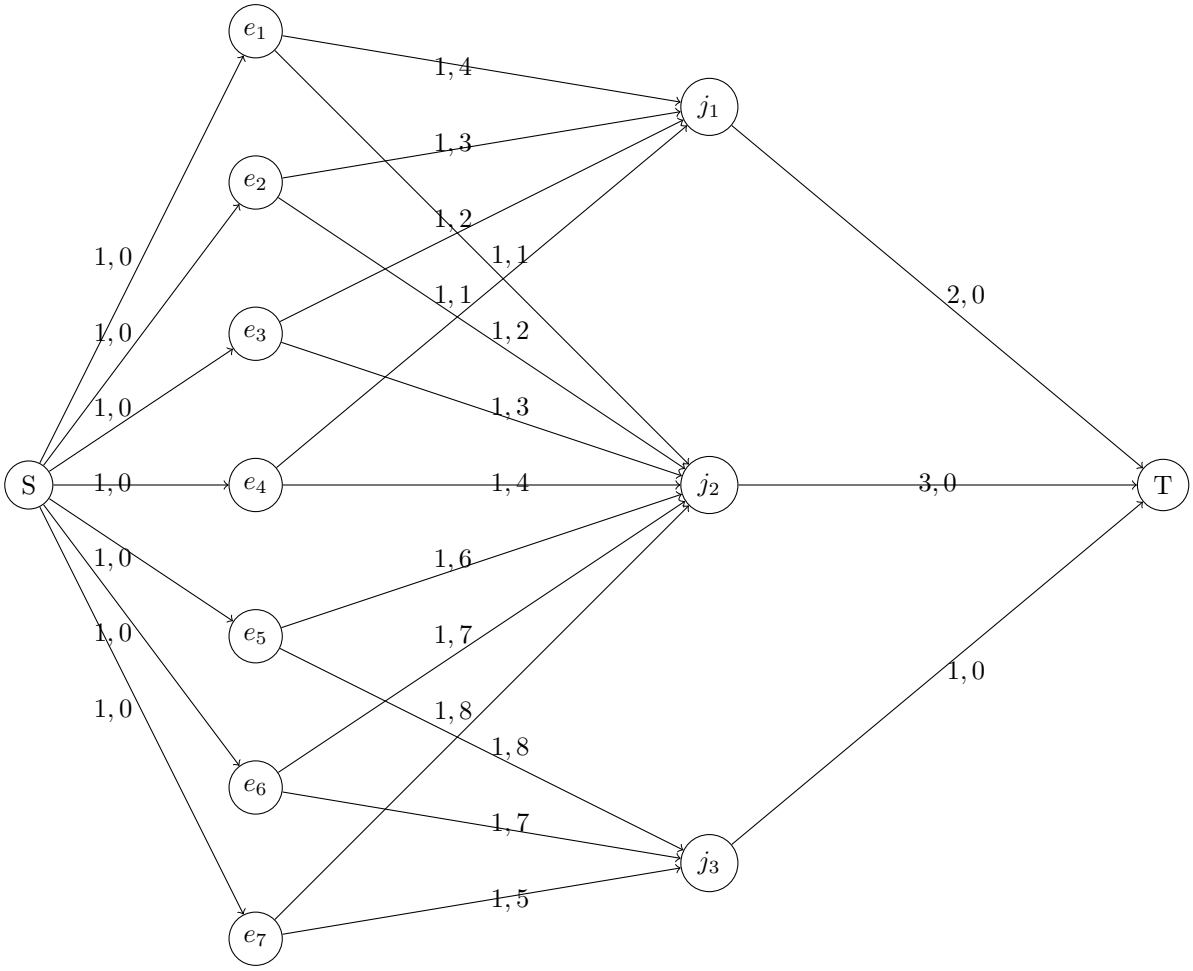
Дугам, якія вядуць з S да супрацоўнікаў, паставім адзінкавыя прапускныя здольнасці і нулявыя кошты; дугам, якія ідуць ад супрацоўнікаў да працаў - адзінкавыя прапускныя здольнасці і кошты c_{ej}; дугам, якія вядуць ад працаў да T - прапускныя здольнасці D_j і нулявыя кошты.

Такім чынам, мы прывялі рашэнне нашай задачы да рашэння задачы пошуку плыні мінімальнага кошту

ў сетцы. Такая задача можа быць эфектыўна вырашана алгарытмам Басакера-Гоўэна: гэта, па сутнасці, алгарытм Форда-Фалкерсана, у якім наяўнасць шляху паміж крыніцай і стокам знаходзіцца не bfs/dfs-ам, а, напрыклад, алгарытмам Форда-Бэлмана, дзе вагі дугаў - кошты.

Прыклад

Разгледзім наступны прыклад задачы. 7 супрацоўнікаў, 3 працы. Першыя чатыры супрацоўнікі кваліфікаваныя рабіць першую альбо другую працу, астатнія тры - другую альбо трэцюю. Першая праца патрабуе два супрацоўніка для выканання, другая - тры, трэцяя - аднаго. Расставім кошты якім-небудзь чынам, атрымліваем наступны граф (сетку):



Пара (x, y) азначае адпаведна прапускную здольнасць і кошт за праход адзінкі.

Сфармавалі сетку - запускаем алгарытм. Атрымліваем вынік:

```
Schedule:
Job 1  by 3
Job 1  by 4
Job 2  by 1
Job 2  by 2
Job 2  by 5
Job 3  by 7
Max flow:  6
Min cost:  17
```

То бок кошт мінімізуецца пры наступным раскладзе:

$j_1 : e_3, e_4$

$j_2 : e_1, e_2, e_5$

$j_3 : e_7$.

Заўважым, што калі значэнне плыні не роўнае суме попытаў усіх працаў, гэта азначае, што расклада не існуе.

Тэкставы файл з уваходнымі дадзенымі для гэтага прыклада і рэалізацыю алгарытма на C++ далучаю асобнымі файламі.