Нижние границы для динамической задачи упаковки в контейнеры с аффинными ограничениями на уровне больших доменов

Турнаев А.М.

Институт математики им. С.Л. Соболева СО РАН (Омский филиал)

Исследование выполнено за счет гранта Российского научного фонда № 22-71-10015-П

28.03.2026

Постановка задачи

1. Имеется потенциально неограниченное количество идентичных подсетей.
2. Каждая подсеть содержит заданное число идентичных серверов (200).
3. Каждый сервер имеет архитектуру неравномерного доступа к памяти (NUMA) с заданным количеством NUMA-узлов (2 или 4).
4. Каждый NUMA-узел имеет ограничения на два ресурса: ЦПУ и память. NUMA-узлы не являются однородными и могут отличаться.

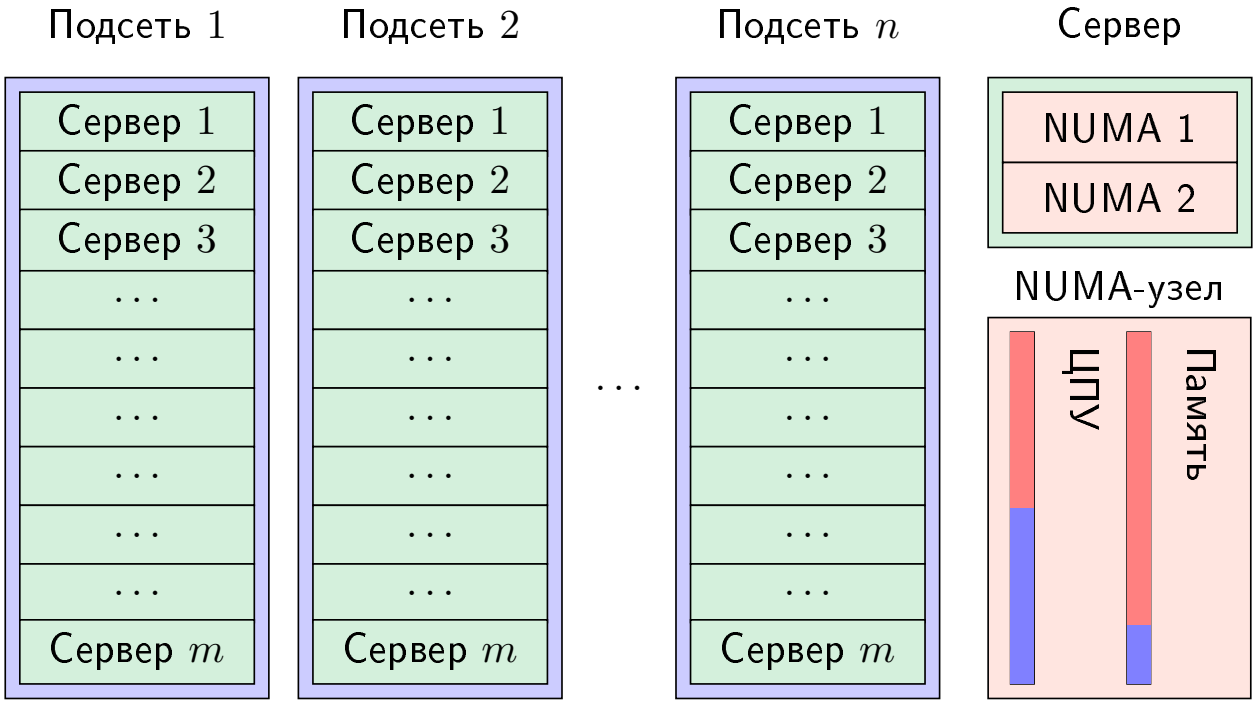

Рис. 1: Структура облачного центра

Постановка задачи

- Для предоставления пользователю вычислительных мощностей, в облачном центре аллоцируются виртуальные машины, то есть размещаются в соответствующей подсети, сервере и NUMA-узле. Виртуальные машины типизированы, а именно тип задает следующие характеристики:
  1. Ресурсные требования – использование ЦПУ и объем оперативной памяти, необходимые для запуска данной виртуальной машины;
  2. Размер – количество частей, из которых состоит данная виртуальная машина. Части виртуальной машины имеют одинаковые ресурсные требования и должны быть размещены на одном сервере, но на разных NUMA-узлах. Мы рассматриваем два размера ВМ: малые (1 часть) и большие (2 части).
- Вычислительные мощности предоставляются на определенные периоды времени, задавая время создания α и время удаления ω для каждой виртуальной машины.

Дополнительные ограничения

- Пользователь может потребовать максимальной производительности. В таком случае виртуальные машины из одной группы должны быть упакованы на одну подсеть, если их времена работы пересекаются. Это минимизирует сетевые задержки.

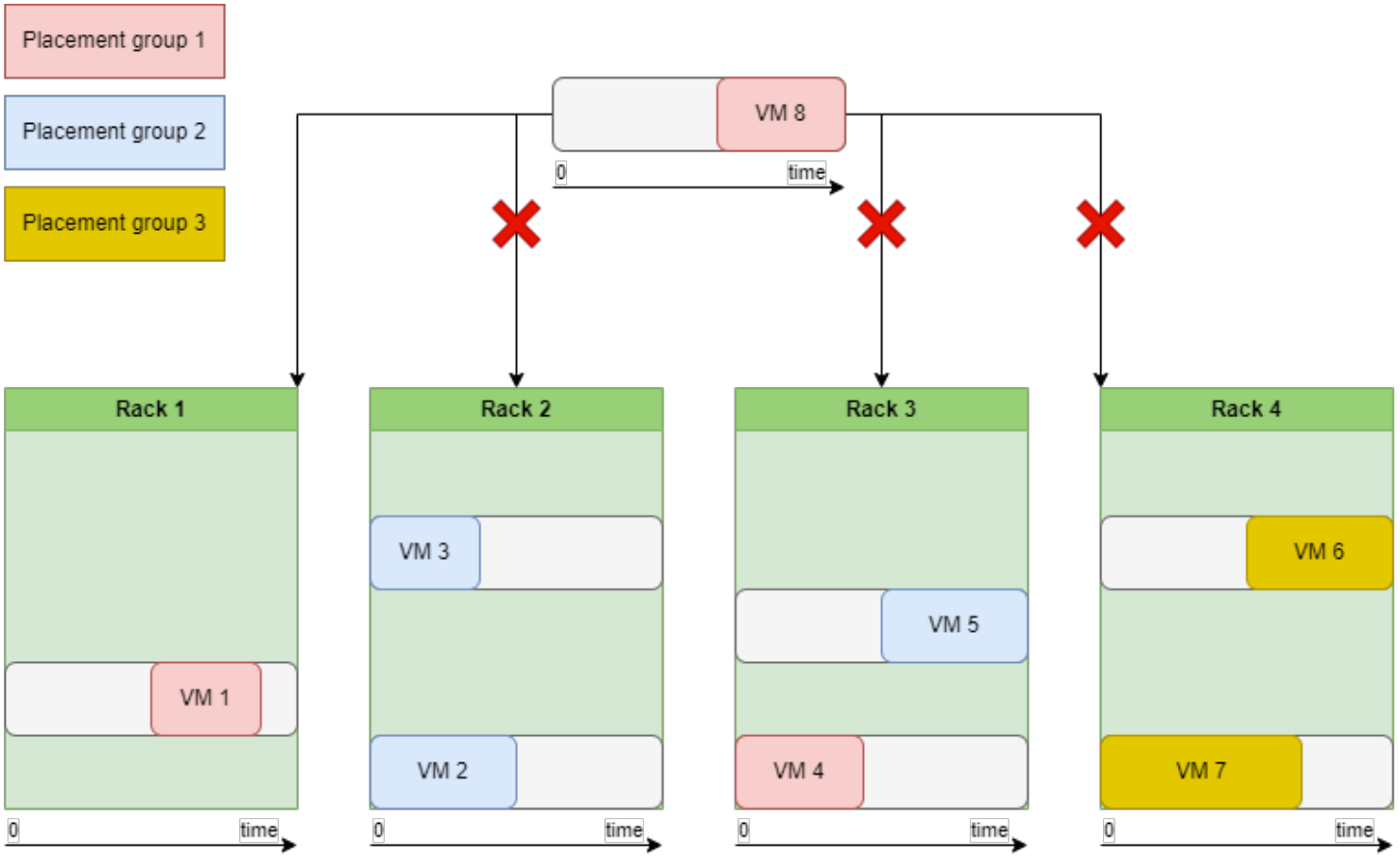

Рис. 2: Иллюстрация аффинных ограничений на размещение

Постановка задачи

Задача

Задача состоит в лексикографической минимизации числа активных подсетей и серверов при заданном наборе запросов на виртуальные машины, с учётом ограничений на ресурсы и размещение.

- Задача является NP-трудной, если количество типов виртуальных машин не ограничено, поскольку это обобщение классической задачи упаковки в контейнеры.
- В работе[1] сказано, что увеличение плотности размещения виртуальных машин даже на 1% позволяет экономить до 100 миллионов долларов в год облачной платформе Microsoft Azure.

[1] Hadary O. et al. Protean:VM allocation service at scale //14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20). 2020. Ñ. 845-861.

Оценки. Свойства задачи

- Будем обозначать количество подсетей как $|N|$, а количество серверов как $|S|$.
- Если $|N_1| > |N_2| \Rightarrow |S_1^*| \leq |S_2^*|$, где $|S_i^*|$ – оптимальное количество серверов, если зафиксировать количество подсетей равным $|N_i|$. Более того, имеются случаи, когда неравенство становится строгим. В связи с этим имеет смысл рассматривать следующие оценки:
  1. $LB_{|N|}^{|S|} \leq |S^*|$, где $|S^*|$ – оптимальное количество серверов, если зафиксировать количество подсетей равным $|N|$;
  2. $LB^{|N|} \leq |N|$ – оценка на допустимое (оптимальное) количество подсетей;
  3. $LB_{|N|}^{K} \leq K$ – оценка на номер последнего активного сервера;
  4. $LB_{|N^*|}^{|S|} \leq |S^*|$ – оценка на количество серверов, если выбрать оптимальное количество подсетей;
  5. $LB_{\infty}^{|S|} \leq |S^*|$ – оценка на количество серверов, без ограничений на количество подсетей.

Нижняя оценка для базовой задачи. Алгоритм генерации столбцов.

Для упрощения рассмотрим классическую задачу упаковки в контейнеры с емкостью контейнера W и весами предметов w_i, $i \in I$ – типы предметов.

- Будем называть произвольный вектор $(a_i)_{i \in I}$ шаблоном упаковки, если он удовлетворяет ограничению: $\sum\limits_{i \in I} a_i \cdot w_i \leq W$.
- Рассмотрим множество всех возможных шаблонов J. Тогда задача упаковки в контейнеры может быть описана как:

$$\begin{cases} \sum\limits_{j \in J} x_j \to \min \\ \sum\limits_{j \in J} a_{ij} x_j \geq n_i, \ i \in I \\ x_j \in \mathbb{Z}_+, \ j \in J \end{cases}$$

Базовая нижняя оценка. Алгоритм генерации столбцов.

- Релаксируем переменные x_j, тогда решение задачи:

$$\begin{cases} \sum\limits_{j \in J} x_j \to \min \\ \sum\limits_{j \in J} a_{ij} x_j \geq n_i, \ i \in I \\ x_j \geq 0, \ j \in J, \end{cases}$$

будет нижней оценкой для исходной задачи, но $|J|$ экспоненциально большое. Поэтому рассмотрим подмножество $J' \subseteq J$, при котором данная задача будет разрешимой(можно получить с помощью жадных алгоритмов FF, BF, ...).

Нижняя оценка для базовой задачи. Алгоритм генерации столбцов.

- Пусть x^* – оптимальное решение для J', а $\lambda^* \geq 0$ – оптимальное решение двойственной задачи для J':

$$\begin{cases} \sum\limits_{i \in I} n_i \lambda_i \to max \\ \sum\limits_{i \in I} a_{ij} \lambda_i \leq 1, \ j \in J' \\ \lambda_i \geq 0, \ i \in I \end{cases}$$

- Если для всех $j \in J$ справедливо:

$$\sum_{i \in I} a_{ij} \lambda_i^* \leq 1,$$

то λ_j^* – оптимальное решение двойственной задачи для всего J, и нижняя оценка сохраняется.

Нижняя оценка для базовой задачи. Алгоритм генерации столбцов.

- Для того чтобы быстро проверить, что:

$$\sum_{i \in I} a_{ij} \lambda_i^* \leq 1, \forall j \in J, \qquad (*)$$

рассмотрим задачу генерации нового шаблона (Pricing Problem):

$$\begin{cases} \alpha = \sum_{i \in I} \lambda_i^* y_i \to \max \\ \sum_{i \in I} w_i y_i \leq W \\ y_i \in \mathbb{Z}_+, \end{cases}$$

Легко заметить, что, если $\alpha^* \leq 1$, то выполняются все ограничения $(*)$, иначе мы получили новый шаблон, который добавляем в J' и повторяем итерацию.

Отсутствие начального допустимого решения

- На самом деле не обязательно иметь начальное допустимое подмножество шаблонов $J' \subseteq J$, его можно построить тем же способом, что и оптимальный набор J^*.
- Задача неразрешима $\Rightarrow$ Двойственная неограничена.
  Пусть $\widehat{\lambda}$ – направляющий вектор экстремального луча.
- Необходимо отсечь "бесконечное" решение, для этого рассмотрим неравенство для любого допустимого решения $\widetilde{\lambda} = s\widehat{\lambda}$:

$$\sum_{i \in I} a_{ji}\widetilde{\lambda}_i \leq 1 \Rightarrow s\sum_{i \in I} a_{ji}\widehat{\lambda}_i \leq 1 \Rightarrow \sum_{i \in I} a_{ji}\widehat{\lambda}_i \leq \frac{1}{s} \xrightarrow{s \to +\infty} \sum_{i \in I} a_{ji}\widehat{\lambda}_i \leq 0$$

- Таким образом, снова достаточно максимизировать $\alpha = \sum\limits_{i \in I} \widehat{\lambda}_i y_i$. Если $\alpha^* \leq 0$, то задача неразрешима для всего J, иначе добавляем новый шаблон.

Базовая нижняя оценка. Алгоритм генерации столбцов.

Алгоритм (Базовой нижней оценки)

1. *Выбрать множество самых нагруженных моментов времени* T.
2. *Вычислить нижние границы в фиксированный момент времени* $t \in T$ *с помощью алгоритма генерации столбцов:* $LB_t = \mathrm{CG}(VM_t)$
3. *Нижняя граница:* $LB = \max\limits_{t \in T}\{LB_t\}$

Кочетов Ю. А., Ратушный А. В. Верхние и нижние оценки оптимума для задачи динамической упаковки в контейнеры //Труды Института математики и механики УрО РАН. – 2024. – Т. 30. – №. 1. – С. 109-127.

Модель ЦЛП №1 (Множества, параметры, переменные)

- Множества:
  1. I – множество типов ВМ;
  2. J – множество статических шаблонов упаковки сервера;
  3. T – множество выбранных моментов времени;
  4. N – множество подсетей;
  5. G – множество групп размещения.
- Параметры:
  1. a_{ji} – количество ВМ типа $i \in I$ в паттерне $j \in J$;
  2. K_{ti} – количество ВМ типа $i \in I$, в момент времени $t \in T$;
  3. K^g_{ti} – множество ВМ типа $i \in I$ из группы $g \in G$, в момент времени $t \in T$;
  4. $\Delta^{\pm}_{ti}$ – множество ВМ типа $i \in I$, созданных (удаленных) в момент времени $t \in T$;
  5. M – количество серверов в одной подсети.
- Переменные:
  1. $x_{ntj} \in \mathbb{Z}_+$ – количество паттернов типа $j \in J$, выбранных для подсети $n \in N$ и момента $t \in T$;
  2. $\delta^+_{nti} \in \mathbb{Z}_+$ – количество ВМ типа $i \in I$, аллоцированных в подсети $n \in N$ в момент времени $t \in T$;
  3. $\delta^-_{nti} \in \mathbb{Z}_+$ – количество ВМ типа $i \in I$, удаленных из подсети $n \in N$ в момент времени $t \in T$;
  4. $c_{gn} \in \{0, 1\}$ равная 1, если для группы $g \in G$ выбрана подсеть $n \in N$;
  5. $s_n \in \mathbb{Z}_+$ – количество активных серверов в подсети $n \in N$.

Модель ЦЛП №1 (Целевая функция, ограничения)

1. Целевая функция (минимизация количества активных серверов):

$$\sum_{n \in N} s_n \to \min$$

2. Ограничения снизу на общее количество упакованных машин:

$$\sum_{n \in N} \sum_{j \in J} a_{ji} x_{ntj} \geq K_{ti}, \ t \in T, i \in I$$

3. Ограничения распределения ВМ, создающихся в момент времени $t \in T$, по подсетям:

$$\sum_{n \in N} \delta^{+}_{nti} = \Delta^{+}_{ti}, \ t \in T, i \in I$$

4. Ограничения распределения ВМ, удаляющихся в момент времени $t \in T$, по подсетям:

$$\sum_{n \in N} \delta^{-}_{nti} = \Delta^{-}_{ti}, \ t \in T, i \in I$$

Модель ЦЛП №1 (Ограничения)

5. Синхронизация создания и удаления виртуальных машин в соседние моменты времени:

$$\sum_{j \in J} a_{ji} x_{ntj} \geq \sum_{j \in J} a_{ji} x_{n(t-1)j} - \delta^-_{nti} + \delta^+_{nti}, \; n \in N, t \in T \setminus \{0\}, i \in I$$

6. Ограничения на назначение группы подсети:

$$\sum_{n \in N} c_{gn} = 1, \; g \in G$$

7. Покрытие групп размещения шаблонами:

$$\sum_{j \in J} a_{ji} x_{ntj} \geq \sum_{g \in G} c_{gn} K_{gti}, \; n \in N, t \in T, i \in I$$

Модель ЦЛП №1 (Ограничения)

8. Ограничения на количество серверов в подсети:

$$\sum_{j \in J} x_{ntj} \leq M, \ n \in N, t \in T$$

9. Ограничения на число активных серверов в n-й подсети:

$$\sum_{j \in J} x_{ntj} \leq s_n, \ t \in T, n \in N$$

10. Ограничения переменных:

$$x_{ntj}, \delta^{\pm}_{nti}, s_n \in \mathbb{Z}_+, \quad c_{gn} \in \{0, 1\}$$

- Данная модель даёт нижнюю границу на количество активных серверов, если зафиксировать мощность множества подсетей $|N|$. Если доказано, что $|N|$ – допустимое количество подсетей, то оценка глобальная.

Декомпозиция Бендерса. Мастер-задача

- Предлагается произвести декомпозицию Бендерса по переменным c_{gn}, и при этом остальные переменные линейно релаксировать:

$$\begin{cases} \theta \to \min \\ \sum\limits_{n \in N} c_{gn} = 1, \ g \in G \\ \text{cuts}(\theta, c) \leq 0 \end{cases}$$

Мастер-задача (MP)

1. Оптимальное решение мастер-задачи $= LB$ (Нижняя граница).
2. Оптимальное решение подзадачи $= UB$ (Верхняя граница).
3. Итеративно решаем, добавляя отсечения, пока $|UB - LB| > \varepsilon$.

Подзадача (двойственная)

$$\sum_{t\in T}\sum_{i\in I}\left(K_{ti}\alpha_{ti}+\Delta^{+}_{ti}\gamma_{ti}+\Delta^{-}_{ti}\varphi_{ti}\right)+\sum_{n\in N}\sum_{t\in T}\sum_{i\in I}\left(\sum_{g\in G}c^{*}_{gn}K_{gti}\right)\mu_{nti}-$$

$$-M\sum_{n\in N}\sum_{t\in T}\pi_{nt}\to\max$$

$$\boxed{\sum_{i\in I}a_{ji}(\alpha_{ti}+\mu_{nti}+\beta_{nti}-\beta_{n(t+1)i})-\pi_{nt}-\rho_{nt}\le 0, n\in N, t\in T, j\in J}$$

$$\gamma_{ti}\le\beta_{nti}, t\in T, n\in N, i\in I$$

$$\beta_{nti}\le-\varphi_{ti}, t\in T, n\in N, i\in I$$

$$\sum_{t\in T}\rho_{nt}\le 1, n\in N$$

$$\alpha,\beta,\gamma,\mu,\pi,\rho\ge 0$$

Двойственная подзадача (DSP)

Отсечения в мастер-задаче

- Отсечение оптимальности:

$$\theta \geq \sum_{t\in T}\sum_{i\in I}\left(K_{ti}\alpha^*_{ti} + \Delta^+_{ti}\gamma^*_{ti} + \Delta^-_{ti}\varphi^*_{ti}\right) + \sum_{n\in N}\sum_{t\in T}\sum_{i\in I}\left(\sum_{g\in G} c_{gn}K_{gti}\right)\mu^*_{nti} - M\sum_{n\in N}\sum_{t\in T}\pi^*_{nt}$$

- Отсечение допустимости:

$$0 \geq \sum_{t\in T}\sum_{i\in I}\left(K_{ti}\alpha^*_{ti} + \Delta^+_{ti}\gamma^*_{ti} + \Delta^-_{ti}\varphi^*_{ti}\right) + \sum_{n\in N}\sum_{t\in T}\sum_{i\in I}\left(\sum_{g\in G} c_{gn}K_{gti}\right)\mu^*_{nti} - M\sum_{n\in N}\sum_{t\in T}\pi^*_{nt}$$

- Далее правую часть будем записывать как:

$$C + \sum_{n\in N}\sum_{g\in G} K'_{ng}c_{ng}$$

Модель ЦЛП №2

- Можно рассмотреть другую модель. Изменим следующее:

1 Целевая функция (Количество активных серверов на всех подсетях $\Rightarrow$ количество активных серверов в последней подсети):

$$\sum_n s_n \to \min \Rightarrow s \to \min$$

8 Ограничения на количество серверов в подсети (убираем ограничение у последней):

$$\sum_{j \in J} x_{ntj} \leq M,\ n \in N, t \in T \Rightarrow \sum_{j \in J} x_{ntj} \leq M,\ n \in N \setminus |N|, t \in T$$

9 Ограничения на число активных серверов в n-й подсети $\Rightarrow$ ограничение на число активных серверов в $|N|$-й подсети:

$$\sum_{j \in J} x_{ntj} \leq s_n,\ t \in T, n \in N \Rightarrow \sum_{j \in J} x_{|N|tj} \leq s,\ t \in T$$

Модель ЦЛП №2. Оценки

- Данная модель даёт границу на количество активных серверов в последней подсети (номер последнего активного сервера).
- Заметим, что, если $s^* \geq M \Rightarrow |N|$ – недопустимое число подсетей.
- Таким образом с помощью декомпозиции Бендерса и генерации столбцов можно оценить снизу:
  1. Минимальное количество подсетей;
  2. Минимальный номер последнего активного сервера;
  3. Минимальное количество активных серверов при зафиксированном числе подсетей;
  4. Минимальное количество серверов, если зарезервировано минимальное количество подсетей (для этого нужно знать допустимое количество подсетей).

Ускорение мастер-задачи

- Для ускорения мастер-задачи можно сделать следующее:
  1. Удаление симметрий связанных с перестановкой подсетей для групп:

$$c_{gn} \leq \sum_{k \in G} c_{k,n-1}, n \in N \setminus \{1\}, g \in G \setminus \{1\}$$

  2. Отбраковка заранее невозможных размещений:

$$\sum_{i \in I} \sum_{g \in G} w_i \cdot K_{gti} c_{gn} \leq W, n \in N, t \in T,$$

где W – суммарная емкость подсети по какому-то ресурсу, а w_i - ресурсные требования ВМ i-го типа.

Ускорение мастер-задачи (трюк с перестановками)

- Однако имеется более интересная модификация. Отсечения симметричны относительно индекса n:

$$\theta \geq C + \sum_{n \in N} \sum_{g \in G} K'_{gn} c_{gn} \Rightarrow \theta \geq C + \sum_{n \in N} \sum_{g \in G} K'_{g\pi(n)} c_{gn}, \forall \pi \in S_{|N|}$$

- Это можно переписать следующим образом:

$$\theta - C \geq \max_{\pi \in S_{|N|}} \sum_{n \in N} \sum_{g \in G} K'_{g\pi(n)} c_{gn} = f(c)$$

- Правая часть является нелинейной функцией от c, но ее можно отлично линеаризовать.

Ускорение мастер-задачи (трюк с перестановками)

- Рассмотрим задачу о назначении для фиксированного c:

$$f(c) = \sum_{n \in N} \sum_{g \in G} \sum_{m \in N} c_{gn} \cdot K'_{gm} \cdot x_{nm} =$$

$$= \sum_{n \in N} \sum_{m \in N} x_{nm} \cdot \left(\sum_{g \in G} c_{gn} \cdot K'_{gm} \right) \to \max$$

$$\sum_{n \in N} x_{nm} = 1, m \in N$$

$$\sum_{m \in N} x_{nm} = 1, n \in N$$

$x_{nm} \geq 0$ (Можно т.к. матрица тотально унимодулярна)

Ускорение мастер-задачи (трюк с перестановками)

- Двойственная для нее:

$$\sum_{n\in N} u_n + \sum_{m\in N} v_m \to \min$$

$$u_n + v_m \geq \sum_{g\in G} c_{gn} K'_{gm},\ n, m \in N$$

$$u, v \in \mathbb{R}$$

- Таким образом:

$$q \in \left\{ \sum_{n\in N} u_n + \sum_{m\in N} v_m \,\middle|\, u_n + v_m \geq \sum_{g\in G} c_{gn} K'_{gm},\ n, m \in N \right\} \Rightarrow q \geq f(c)$$

- Более того:

$$\min \left\{ \sum_{n\in N} u_n + \sum_{m\in N} v_m \,\middle|\, u_n + v_m \geq \sum_{g\in G} c_{gn} K'_{gm},\ n, m \in N \right\} = f(c)$$

Ускорение мастер-задачи (трюк с перестановками)

- Таким образом ограничение:

$$\theta - C \geq \max_{\pi \in S_{|N|}} \sum_{n \in N} \sum_{g \in G} K'_{g\pi(n)} c_{gn} = f(c)$$

- Можно заменить на:

$$\theta \geq C + \sum_{n \in N} u_n + \sum_{m \in N} v_m$$

$$u_n + v_m \geq \sum_{g} K'_{gm} c_{gn}, \ n, m \in N$$

$$u, v \in \mathbb{R}$$

- Это ускоряет сходимость примерно в 50 раз (время в 20).

Численные эксперименты

- Алгоритмы были проверены на множестве разнородных синтетических тестов. Использовался решатель Gurobi. Некоторые результаты:
  1. Тесты, содержащие группы, которые резервируют от 50% до 70% подсети хотя бы в один момент времени.
     1. Решаются быстро (1-2 минуты);
     2. Дают ощутимое улучшение нижней границы. Количество подсетей увеличивается на 1-4;
     3. Зависимости скорости работы от количества выбранных моментов времени практически нет.
     4. Подмешивание малых групп $(10\% - 30\%)$ лишь незначительно сказывается на итоговом времени работы, при этом нижняя граница увеличивается.
  2. Тесты, содержащие группы, которые резервируют от 30% до 50% подсети хотя бы в один момент времени.
     1. Решаются долго (10-60 минут);
     2. Дают различное улучшение нижней границы.
     3. Сильная зависимость от количества выбранных моментов времени.
  3. Тесты, содержащие группы, которые резервируют от 10% до 30% подсети хотя бы в один момент времени.
     1. Не поддаются алгоритму;
     2. При релаксации части групп размещения (самых малых) решение находится, но не улучшает нижнюю границу;
     3. По всей видимости модель в больше степени учитывает невозможность упаковки вместе нескольких групп, чем требование размещения всех ВМ из одной группы на одну подсеть.

Численные эксперименты

- В основном алгоритмы тратят больше всего времени на решение мастер-задачи. При этом итоговое количество итераций редко превышает 50, а модели мастер-задачи получаются небольшими (500 бинарных переменных, 300×300 матрица).
- Модель №1 решается значительно хуже, чем модель №2, вплоть до невозможности дождаться улучшения нижней границы;

Численные эксперименты

name	$\|G\|$	$\|VM\|$	UB_N	UB_K	US_S	LB_S	LB_N	LB_K	$LB_S^{\|UB_N\|}$	t
test1	116	103360	8	1491	1491	1185	7	1363	1243	10m
test2	132	40407	4	705	705	634	4	675	675	60s

Таблица 1: Данные экспериментов

Заключение

1. Был разработан комплексный алгоритм оценки нижних границ для динамической задачи упаковки в контейнеры с афинными ограничениями на уровне больших доменов;
2. Алгоритм даёт хорошие результаты, но на специфических примерах – сложных для решения, но лёгких для оценки;
3. Можно попробовать решать мастер-задачу неоптимально: есть вероятность, что неоптимальные отсечения ускорят сходимость;
4. Возможно, стоит отказаться от некоторых ограничений в изначальной модели (3,4,5), поскольку, по-видимому, на итоговую оценку влияет именно невозможность разместить несколько групп на одной подсети;
5. Необходимы дальнейшие эксперименты, связанные с генератором тестов, для обнаружения предела применимости.

Спасибо за внимание!