Применение методов линейного программирования к нелинейной постановке задачи бесконфликтного слияния потоков воздушных судов

А.А. Спиридонов

12 декабря 2019 г.

Постановка задачи

- $t_i^{\text{nom}} \to [t_i^{\text{nom}} - t_i^{\text{acc}}, t_i^{\text{nom}} + t_i^{\text{dec}}]$ — интервал варьирования изменённого момента прибытия каждого воздушного судна (ВС).
  В моделировании использовался единый для каждого ВС интервал варьирования $[t^{\text{acc}}, t^{\text{dec}}]$;
- $\{t_i^{\text{nom}}\}_{i=1}^{N}$ — упорядоченный по возрастанию набор номинальных моментов прибытия ВС в точку слияния;
- τ_{safe} — минимальный безопасный временной интервал между двумя ВС в упорядочиваемой очереди.

Постановка задачи

Таким образом, возникает оптимизационная задача

$$F(\{t_i\}, \{t_i^{\mathsf{nom}}\}) = \sum_{i=1}^{N} f(t_i, t_i^{\mathsf{nom}}) \to \min$$

при ограничениях

$$t_i \in [t_i^{\mathsf{nom}} - t_i^{\mathsf{acc}}, t_i^{\mathsf{nom}} + t_i^{\mathsf{dec}}], \quad t_i \geqslant 0,$$
$$\forall\, 1 \leqslant i < j \leqslant N \quad |t_i - t_j| \geqslant \tau_{\mathsf{safe}}$$

Задача безусловной минимизации

Жёсткие ограничения, связанные с интервалом варьирования момента прибытия:

$$-\alpha \cdot \ln(t_i - t_i^{\mathsf{nom}} + t^{\mathsf{acc}}), \quad -\alpha \cdot \ln(-t_i - t_i^{\mathsf{nom}} + t^{\mathsf{dec}});$$

Функционал для безусловной минимизации:

$$\mathcal{F} = F(\{t_i\}, \{t_i^{\mathsf{nom}}\}) - \alpha \cdot \sum_{i=1}^{N} \Big(\ln(t_i - t_i^{\mathsf{nom}} + t^{\mathsf{acc}}) + + \ln(-t_i - t_i^{\mathsf{nom}} + t^{\mathsf{dec}}) \Big) + \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} s(t_i, t_j)$$

Функция $s(t_i, t_j)$ штрафует взаимное расположение i-го и j-го судов, например, при отсутствии между ними безопасного временно́го интервала.

Квадратичный критерий

Ранее исследовались кусочно-линейные критерии.

Представляют интерес нелинейные критерии. Однако при решении нелинейных оптимизационных задач возникает ряд сложностей.

Квадратичный критерий

Ранее исследовались кусочно-линейные критерии.

Представляют интерес нелинейные критерии. Однако при решении нелинейных оптимизационных задач возникает ряд сложностей.

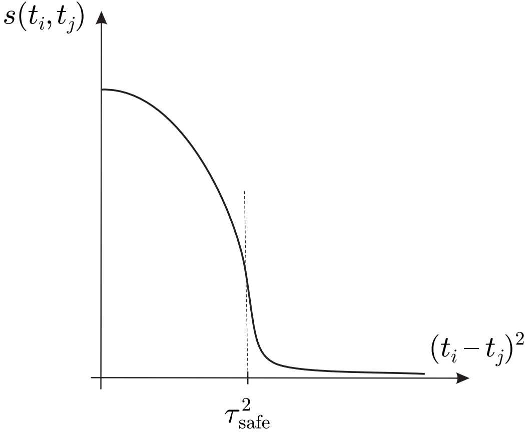
Отработка подходов к решению нелинейных задач велась на простейшем нелинейном критерии вида:

$$F(\{t_i\}, \{t_i^{\mathsf{nom}}\}) = \sum_{i=1}^{N} (t_i - t_i^{\mathsf{nom}})^2 \to \min$$

Доказано утверждение, что порядок воздушных судов на оптимальном решении совпадает с порядком в исходной очереди.

Учёт интервала безопасности

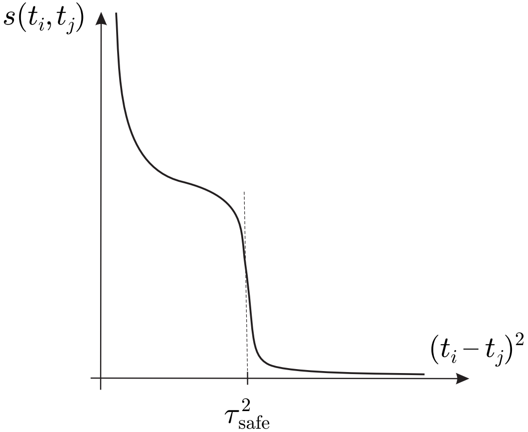
Мягкие ограничения, совпадения номинальных моментов допускаются



В качестве начальной точки берётся набор исходных или скорректированных номинальных моментов прибытия.

Учёт интервала безопасности

Мягкие ограничения, совпадения номинальных моментов ***не*** допускаются



В качестве начальной точки берётся набор ***скорректированных*** номинальных моментов прибытия.

Жёсткие ограничения. Получение внутренней точки

$$s(t_i, t_j) = -\alpha \cdot \ln\left((t_i - t_j)^2 - \tau_{\mathsf{safe}}^2\right).$$

Жёсткие ограничения. Получение внутренней точки

$$s(t_i, t_j) = -\alpha \cdot \ln\left((t_i - t_j)^2 - \tau_{\text{safe}}^2\right).$$

1. Для получения точки, с которой будет стартовать численный метод, используется «жадный» алгоритм. Все воздушные суда были успешно вписаны и нет новых времён прибытия на правых границах $t_i^{\text{nom}} + t^{\text{dec}}$ возможных интервалов прибытия — точка удовлетворяет ограничением, но лежит на границе области (первое судно будет сдвинуто на левую границу своего интервала).
Иначе внутренней точки не существует.

Жёсткие ограничения. Получение внутренней точки

$$s(t_i, t_j) = -\alpha \cdot \ln\left((t_i - t_j)^2 - \tau^2_{\text{safe}}\right).$$

1. Для получения точки, с которой будет стартовать численный метод, используется «жадный» алгоритм. Все воздушные суда были успешно вписаны и нет новых времён прибытия на правых границах $t_i^{\text{nom}} + t^{\text{dec}}$ возможных интервалов прибытия — точка удовлетворяет ограничением, но лежит на границе области (первое судно будет сдвинуто на левую границу своего интервала).
Иначе внутренней точки не существует.
2. В качестве начальной точки берутся «раздвинутые» результаты жадного алгоритма.

Применявшиеся методы нелинейной оптимизации

Применявшиеся методы нелинейной оптимизации

- Метод прямого поиска Хука–Дживса
  Опирается только на значения функции, производные не используются.

Применявшиеся методы нелинейной оптимизации

- Метод прямого поиска Хука–Дживса
  Опирается только на значения функции, производные не используются.
- Метод Ньютона
  Ищет минимум для выпуклых функций, однако в задаче с мягкими ограничениями существуют области вогнутости. В этом случае работает градиентный спуск.

Применявшиеся методы нелинейной оптимизации

- Метод прямого поиска Хука – Дживса
  Опирается только на значения функции, производные не используются.
- Метод Ньютона
  Ищет минимум для выпуклых функций, однако в задаче с мягкими ограничениями существуют области вогнутости. В этом случае работает градиентный спуск.
- Метод Successive Linear Programming

Метод SLP

Алгоритм ориентирован на задачи следующего типа:

$$\begin{aligned} & f(x) \to \min \\ & g_j(x) \leqslant 0, j = 1, \ldots, m_i, \\ & h_r(x) = 0, r = 1, \ldots, m_e, \\ & x \in R^n. \end{aligned} \tag{1}$$

На каждой итерации нелинейного программирования задача (1) локально линеаризуется, после чего в точке $x^{(i)}$ решается задача линейного программирования.

Метод SLP

Задача линейного программирования в точке $x^{(i)}$:

$$\min \nabla f(x^{(i)})^T d + C\left(\sum_{j=1}^{m_i} t_j^g + \sum_{r=1}^{m_e} t_r^{h^+} + \sum_{r=1}^{m_e} t_r^{h^-}\right) \tag{2}$$

$$g_j(x^{(i)}) + \nabla g_j(x^{(i)})^T d \leqslant t_j^g, j = 1, \ldots, m_i \tag{3}$$

$$h_r(x^{(i)}) + \nabla h_r(x^{(i)})^T d = t_r^{h^+} - t_r^{h^-}, r = 1, \ldots, m_e \tag{4}$$

$$(d^{(r)})^T H^{(i)} d = 0, r = 1, \ldots, i-1; i > 1, \tag{5}$$

$$d^L \leqslant d \leqslant d^U, \tag{6}$$

$$0 \leqslant t_j^g \leqslant \max\{g_j\{(x^{(i)}), 0\}, j = 1, \ldots, m_e, \tag{7}$$

$$0 \leqslant t_r^{h^+} \leqslant |h_r(x^{(i)})|, r = 1, \ldots, m_e, \tag{8}$$

$$0 \leqslant t_r^{h^-} \leqslant |h_r(x^{(i)})|, r = 1, \ldots, m_e. \tag{9}$$

Здесь $(d^{(i)}, t^{g,(i)}, t^{h^+,(i)}, t^{h^-,(i)})$ – решение задачи линейного программирования в точке $x^{(i)}$.

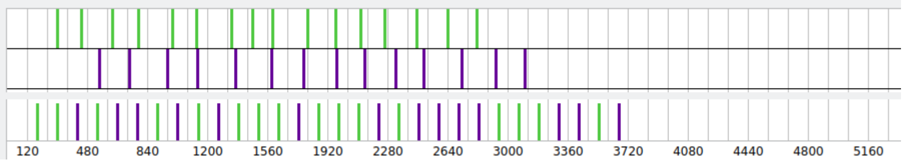
Метод SLP

После того, как на очередной линейной итерации алгоритма решена задача (2) – (9) и, тем самым, выбрано направление d уменьшения функции, из точки $x^{(i)}$ в направлении d производится линейный поиск.

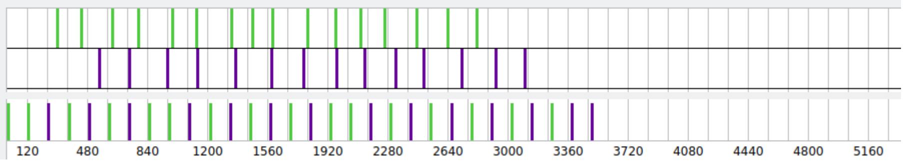
Моделирования очереди из 30 судов

Метод Хука – Дживса

а) мягкие ограничения: порядок ВС был изменён



б) жёсткие ограничения

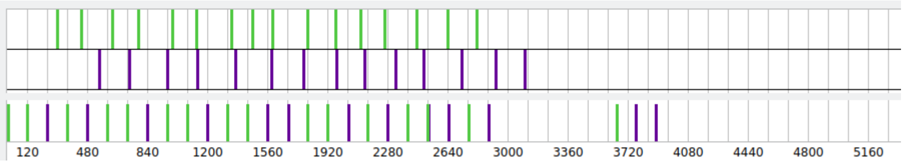


$\tau_{\mathsf{safe}} = 120$ с, $t^{\mathsf{acc}} = 360$ с, $t^{\mathsf{dec}} = 840$ с

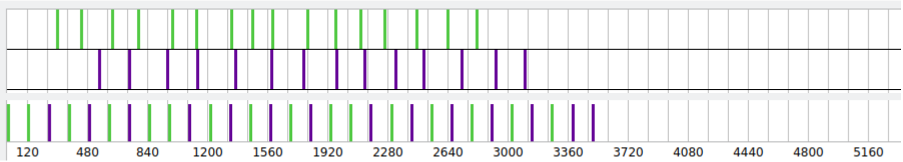
Моделирования очереди из 30 судов

Метод Ньютона

а) мягкие ограничения: **не выдержан** безопасный интервал, превышение максимального количества итераций



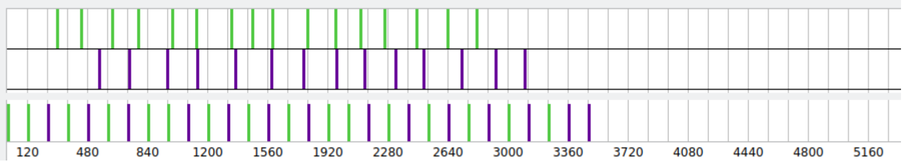
б) жёсткие ограничения: превышение максимального количества итераций



$\tau_{\text{safe}} = 120$ с, $t^{\text{acc}} = 360$ с, $t^{\text{dec}} = 840$ с

Моделирования очереди из 30 судов

Метод SLP

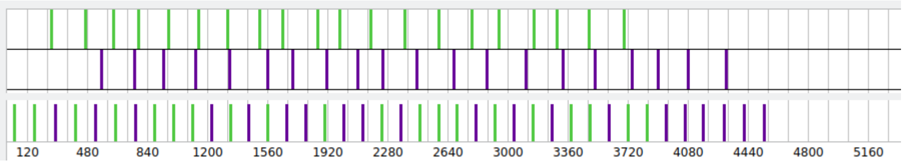


$\tau_{\text{safe}} = 120$ с, $t^{\text{acc}} = 360$ с, $t^{\text{dec}} = 840$ с

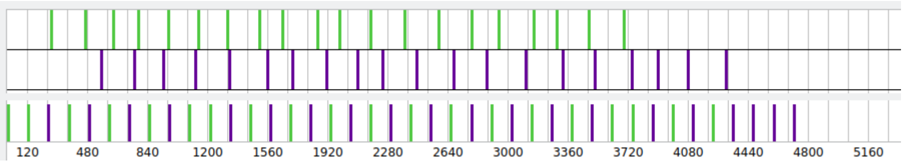
Моделирования очереди из 40 судов

Метод Хука – Дживса

а) мягкие ограничения: порядок ВС был изменён, превышение максимального количества итераций



б) жёсткие ограничения

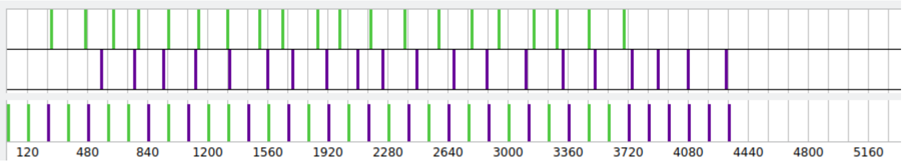


$\tau_{\text{safe}} = 120$ с, $t^{\text{acc}} = 360$ с, $t^{\text{dec}} = 840$ с

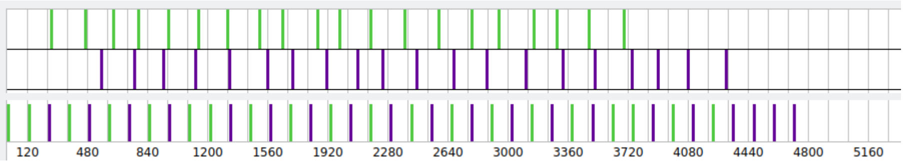
Моделирования очереди из 40 судов

Метод Ньютона

а) мягкие ограничения: **не выдержан** безопасный интервал, порядок ВС был изменён



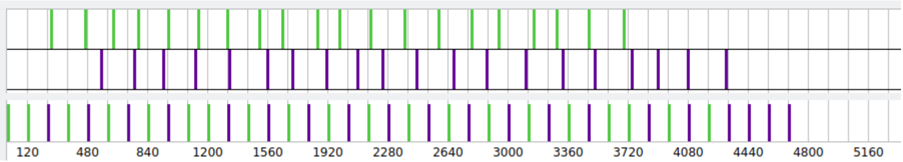
б) жёсткие ограничения: превышение максимального количества итераций



$\tau_{\text{safe}} = 120$ с, $t^{\text{acc}} = 360$ с, $t^{\text{dec}} = 840$ с

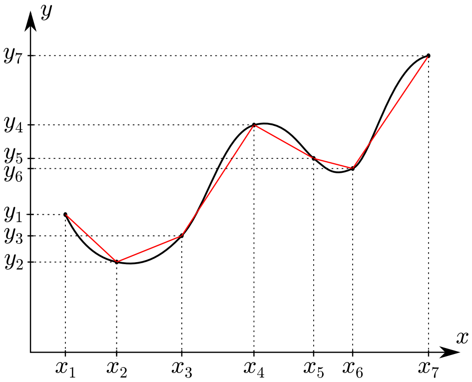
Моделирования очереди из 40 судов

Метод SLP



$\tau_{\text{safe}} = 120$ с, $t^{\text{acc}} = 360$ с, $t^{\text{dec}} = 840$ с

Кусочно-линейная аппроксимация



Чёрным обозначена исходная функция, красным — аппроксимация

Кусочно-линейная аппроксимация

Кусочно-линейная функция представима в виде:

$$f(x) = \sum_{i=2}^{n-1} a_i |x - x_i| + Ax + B$$

После суммирования кусочно-линейных аппроксимаций критериев для каждого судна получаем кусочно-линейный общий функционал, вообще говоря, невыпуклый.

Возможности дальнейшей работы с ним:

- искусственный переход к задаче линейного программирования;
- овыпуклениe функционала и решение эквивалентной задачи ЛП;
- использование методов кусочно-линейного программирования.

Кусочно-линейная аппроксимация. Переход к задаче линейного программирования

Для применения методов линейного программирования требуется переход к задаче линейного программирования:

$$\xi_i := |x - x_i|,\ i = 2, \ldots, n-1$$

$$f(x) = \sum_{i=2}^{n-1} a_i \xi_i + Ax \to \min$$

$$-\xi_i \leqslant x - x_i \leqslant \xi_i$$

Проблема: ξ_i — лишь оценка сверху на значения $|x - x_i|$. Те ξ_i, которые соответствуют отрицательным a_i в процессе работы алгоритма неограниченно возрастают и через них находится требуемое значение x.

Кусочно-линейная аппроксимация. Переход к выпуклой функции

Если кусочно-линейная функция такова, что возрастает неограниченно при значении аргумента, стремящемся к бесконечности, то её минимум равен минимуму её овыпукления. Невыпуклый функционал, получаемый суммированием невыпуклых штрафов каждого ВС, можно овыпуклить и решать задачу линейного программирования.

Основной сложностью на этом пути является отсутствие устойчивой процедуры многомерного овыпукления. Поиск или разработка таких алгоритмов являются отдельным исследованием.

Кусочно-линейное программирование

Другой подход к минимизации рассматриваемой функции может заключаться в использовании методов кусочно-линейного программирования. Теоретическое рассмотрение этого раздела математики и связи его с линейным программированием проводились в Екатеринбурге в ИММ УрО РАН И.И. Ерёминым.

Однако в его работах содержатся только теоретические исследования и не предлагаются эффективные конструктивные алгоритмы решения кусочно-линейной задачи. Поиск таких алгоритмов или их разработка составляют отдельную проблему.

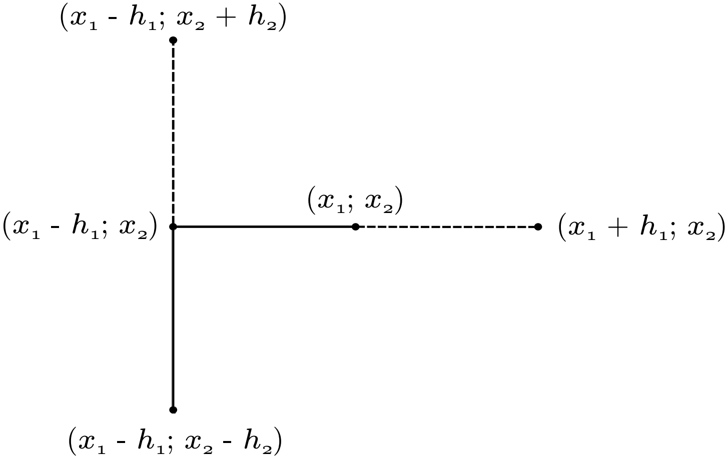
Заключение

Выполнен анализ работы методов Хука–Дживса, метода Ньютона и метода SLP для простейшего нелинейного критерия — квадратичного — при жёстких и мягких ограничениях.

Дальнейшая работа

1. Рассмотреть применение метода SLP к другим функционалам качества слитой очереди.
2. Изучить работу метода SLP на задачах с неодинаковыми ограничениями на промежуток безопасности за разными ВС и с неодинаковыми промежутками варьирования у разных судов.
3. В случае возникновения ограничений, описываемых нелинейными функциями, изучить вопрос эффективности алгоритма SLP.
4. Изучить возможность использования кусочно-линейных аппроксимаций нелинейных критериев.

Метод Хука – Дживса



Метод Хука – Дживса