Оглавление

Глава 1

NP задачи

1.1 Экстремальные задачи

Определение 1. NP-полные задачи оптимизации называются NP-трудными

Классификация приближённых алгоритмов.

1. Жадные алгоритмы
2. Алгоритмы с гарантированной оценкой точности
3. Приближённые алгоритмы, которые выдают одно решение
4. Метаэвристики:
   (a) Поиск в локальной окрестности:
      i. Имитиация отжига
   (b) Генетические алгоритмы
   (c) Муравбиный поиск
   (d) Вероятностные алгоритмы

1.1.1 Жадные алгоритмы

Жадный алгоритм для задачи коммивояжёра – идти в ближайший город

1.1.2 Гарантированная оценка точности

Оценка существует:

$$\exists\, k : \forall \text{ решения } A \quad \frac{f_A}{f_{\text{опт}}} \le k$$

Оценки не существует:

$$\forall k \quad \exists \text{ решение } A : \frac{f_A}{f_{\text{опт}}} > k$$

Следующий алгоритм работает только на неориентированном графе:

Алгоритм (Эйлера).

1. Строим кратчайшее остовное дерево T
2. Удваиваем рёбра дерева. Получаем $2T$
3. Строим эйлеров цикл C
4. Из C делаем гамильтонов цикл (маршрут коммивояжёра) L:
   (a) Вычёркиваем повторы

Теорема 1. Для метрической задачи коммивояжёра, для алгоритма Эйлера верно $\dfrac{f_A}{f_{\text{опт}}} \le 2$

Доказательство.

$$|T| = \sum_{u \in T} l\,\{\,u\,\}\,, \qquad |C| = 2|T|, \qquad f_A = |L| \le 2|T|, \qquad f_{\text{опт}} \ge |T|$$

□

Алгоритм (Кристофидиса).

1. Строим кратчайшее остовное дерево T
2. (a) В построенном дереве выделяем вершины нечётной степени. Их чётное число
   (b) Находим полное паросочетание на этих вершинах с минимальной суммой. Его добавляем к T
3. Строим эйлеров цикл C
4. Из C делаем гамильтонов цикл (маршрут коммивояжёра) L:
   (a) Вычёркиваем повторы

Теорема 2. Алгоритм Кристофидиса на метрической задаче имеет гарантированную оценку $\dfrac{3}{2}$

1.1.3 Метаэвристики

Поиск в локальной окрестности (LS – Local Search)

X – решение, $U(X)$ – окрестность

Определение 2. Окрестность определяется следующим образом:
Задаётся набор операций над X. $U(X)$ – все решения, которые можно получить из X этими операциями

Остановка:

- По числу итераций
- По времени
- Если целевая функция какое-то количество итераций (времени) не уменьшается
- Отклонение f от нижней оценки меньше заданного

Можно время от времени переходить в плохое решение:
Если встретили плохое решение, подбрасываем монетку ($\approx 90\%$ и 10%). Если выпало 10%, переходим в это решение

Генетические алгоритмы

X – особь, у которой есть генотип и фенотип
Фенотип – $f(X)$ – значение целевой функции
Генотип – *что-то*, например, перестановка
Заранее определяем n – размер популяции

Алгоритм.

1. Разделяем популяцию на пары. Получаем $n/2$ пар
2. От каждой пары получаем 2 потомка. Операция – кроссинговер (кроссовер):

(a) "Комбинируем" перестановки:

$$\begin{matrix} a_1 & ... & a_{k/2} & | & a_{k/2+1} & ... & a_k \\ b_1 & ... & b_{k/2} & | & b_{k/2+1} & ... & a_k \end{matrix} \implies \begin{matrix} a_1 & ... & a_{k/2} & | & b_{k/2+1} & ... & b_k \\ b_1 & ... & b_{k/2} & | & a_{k/2+1} & ... & a_k \end{matrix}$$

3. Отбор n лучших особей

Возможные улучшения.

- Двухточечный кроссовер
- При отборе оставлять несколько плохих особей
- Более сильная особь оставляет больше потомства:
  1. Каждой особи приписыаем коэффициент качества
  2. Вместо кроссовера $n/2$ раз запускаем схему Уолкера и решаем, кто будет размножаться