ДОКЛАДЫ АКАДЕМИИ НАУК, 2013, том 450, № 4, с. 393–396

ИНФОРМАТИКА

УДК 519.1

ЭФФЕКТИВНОЕ ВЫЧИСЛЕНИЕ ДОПУСКОВ В ЗАДАЧЕ О ВЗВЕШЕННОМ НЕЗАВИСИМОМ МНОЖЕСТВЕ ДЛЯ ДЕРЕВЬЕВ

© 2013 г. Б. И. Гольденгорин, Д. С. Малышев, П. М. Пардалос

Представлено академиком Ю.Г. Евтушенко 19.12.2012 г.

Поступило 19.12.2012 г.

DOI: 10.7868/S0869565213160032

После получения оптимального решения задачи комбинаторной оптимизации (ЗКО) следующим естественным шагом является анализ его чувствительности. Целью анализа чувствительности полученного оптимального решения ЗКО является выяснение зависимости этого решения от изменения исходных данных. Известны несколько причин интереса к анализу чувствительности. Во-первых, во многих случаях исходные данные ЗКО заданы неточно или имеют естественную неопределенность. Тогда анализ чувствительности необходим для определения доверия как к самому оптимальному решению, так и к выводам, основанным на этом решении. Во-вторых, формальное моделирование существенных свойств искомого оптимального решения в терминах ЗКО часто является трудноформализуемым процессом. Тогда, имея оптимальное решение упрощенной ЗКО, лицо, принимающее решение, интересуется тем, насколько найденное оптимальное решение удовлетворяет свойствам, неучтенным при построении математической модели для упрощенной задачи. Целью изучения таких возмущений является нахождение так называемых допусков, определяемых как максимальное изменение индивидуальной стоимости (веса, времени и т.п.), сохраняющее оптимальность заданного решения, при условии сохранения неизменными остальных данных ЗКО.

Допуски представляют интерес в связи с тем, что минимальное значение допусков элементов оптимального решения задачи является оценкой снизу радиуса устойчивости оптимального решения и служит основой для построения переборных алгоритмов решения различных задач ЗКО [6]. Превышение же максимального значения допусков гарантирует неустойчивость заданного оптимального решения. Первое неявное алгоритмическое использование допусков можно найти в методе Фогеля [8] для нахождения наиболее близкого к оптимальному базисному решению в симплекс-методе решения транспортной задачи, а также в работе [2] при построении эффективной эвристики для решения трехиндексной задачи о назначении. Допуски успешно использовались для улучшения алгоритмов решения не только для “труднорешаемых” задач, но и для эффективно решаемых задач, например в задаче о покрытии графа непересекающимися по вершинам циклами оптимального суммарного веса, известной как линейная задача о назначении [1], а также для многих ее вариаций и обобщений [3]. Особое внимание уделяется эффективно решаемым классам ЗКО, для которых возможно одновременное вычисление оптимального решения и всех допусков. Примерами таких задач являются задача нахождения остовного дерева минимального веса [10], кратчайшего пути [7, 9] и задача о назначении [12].

Данная работа посвящена эффективному определению всех допусков в задаче о взвешенном независимом множестве (ЗВНМ) для деревьев на основе полученной формулы для допусков и динамического программирования. Именно, для дерева с n вершинами данная процедура использует $O(n)$ арифметических операций над вещественными числами произвольной длины. Выбор деревьев в качестве объекта исследований определился тем, что этот класс ЗВНМ имеет невысокую трудоемкость и, следовательно, будет полезен при разработке переборных алгоритмов типа ветвей и границ для решения ЗВНМ в общем случае (см., например, соответствующий выбор для задачи коммивояжера в [5, 11]).

Нижегородский филиал Национального исследовательского университета “Высшая школа экономики”

Нижегородский государственный университет им. Н.И. Лобачевского

Center of Applied Optimization, University of Florida, Gainesville, USA

1. ЗАДАЧА КОМБИНАТОРНОЙ ОПТИМИЗАЦИИ

Задача комбинаторной оптимизации определяется четверкой параметров $\{\Gamma, c, F, f_c\}$, где $c: \Gamma \to R$ – весовая функция универсального множества Γ, F – множество допустимых решений задачи, f_c – целевая функция. Для заданной конкретной четверки $\{\Gamma, c, F, f_c\}$ соответствующая ЗКО состоит в том, чтобы отыскать такой элемент $S^* \in F$, на котором целевая функция принимает свое экстремальное (минимальное или максимальное) значение. Такое множество S^* называется оптимальным решением задачи.

Примером ЗКО является задача о взвешенном независимом множестве. Данная задача для взвешенного по вершинам обыкновенного графа состоит в том, чтобы найти в данном графе множество попарно несмежных вершин наибольшего веса. Напомним, что любое множество попарно несмежных вершин принято называть независимым. Для данной задачи: Γ – множество вершин входного графа, F – семейство независимых множеств входного графа, c – неотрицательная весовая функция вершин, $f_c(S)$ – вес независимого множества S, равный сумме $\sum_{s \in S} c(s)$.

Пусть задана ЗКО на максимум, некоторое ее оптимальное решение S^* и элементы x и y. Данную задачу будем называть старой. По старой задаче образуем две новые задачи. Первая новая задача получается из старой путем уменьшения веса элемента x на неотрицательное число w. Вторая задача получается из старой путем увеличения веса элемента y на некоторое неотрицательное число. Вернемся к рассмотрению старой задачи. Нижним допуском элемента x относительно заданного оптимального решения S^* (обозначаемым через $l_{S^*}(x)$) называется супремум тех чисел w, для которых множество S^* остается оптимальным решением для первой новой задачи, при условии сохранения всех остальных данных старой задачи неизменными. Смысл нижнего допуска состоит в том, что его минимальное значение, найденное среди всех элементов оптимального решения, является оценкой радиуса устойчивости оптимального решения старой задачи. Иными словами, для этой задачи нет допустимого решения S', для которого выполнено неравенство $f_c(S^*) > f_c(S') > f_c(S^*) - l$, где l – минимальный нижний допуск, найденный среди всех нижних допусков элементов из S^*. По аналогии с понятием нижнего допуска можно ввести понятие верхнего допуска. Верхним допуском элемента y относительно оптимального решения S^* (обозначаемым через $u_{S^*}(y)$) называется супремум тех чисел w, для которых множество S^* остается оптимальным решением для второй новой задачи, при условии сохранения всех остальных данных старой задачи неизменными. Смысл верхнего допуска аналогичен смыслу нижнего допуска. Именно, для старой задачи нет допустимого решения S'', для которого выполнено неравенство $f_c(S^*) > f_c(S'') > f_c(S^*) - u$, где u – минимальный из допусков элементов, не принадлежащих S^*.

2. ФОРМУЛА ДЛЯ ВЫЧИСЛЕНИЯ ДОПУСКОВ ДЛЯ ЗАДАЧИ О ВЗВЕШЕННОМ НЕЗАВИСИМОМ МНОЖЕСТВЕ

Далее везде предполагается, что $\{\Gamma, c, F, f_c\}$ – задача о взвешенном независимом множестве. Через $F_-(x)$ будем обозначать совокупность независимых множеств, не содержащих вершину x. Через $F_+(y)$ будем обозначать совокупность независимых множеств, содержащих вершину y. Совокупность независимых множеств, имеющих среди элементов из $F_-(x)$ и $F_+(y)$ наибольший вес, будем обозначать через $F_-^*(x)$ и $F_+^*(y)$ соответственно. Множество F^* – совокупность всех оптимальных решений задачи $\{\Gamma, c, F, f_c\}$. Для подмножества $F' \subseteq F$ через $f_c(F')$ будем обозначать наибольший из весов независимых множеств, принадлежащих F'. Справедлива следующая

Лемма 1. *Пусть $S^* \in F^*$, $x \in S^*$ и $y \notin S^*$. Тогда справедливы равенства*:

а) $l_{S^*}(x) = f_c(F^*) - f_c(F_-^*(x))$;

б) $u_{S^*}(y) = f_c(F^*) - f_c(F_+^*(y))$.

Из леммы 1 следует, что нижние и верхние допуски являются инвариантами множества оптимальных решений в том смысле, что их значения не зависят от выбранного оптимального решения.

3. ВЫЧИСЛЕНИЕ ДОПУСКОВ В ЗАДАЧЕ О ВЗВЕШЕННОМ НЕЗАВИСИМОМ МНОЖЕСТВЕ ДЛЯ ДЕРЕВЬЕВ

Согласно лемме 1, для любых $S^* \in F^*$, $x \in S^*$ и $y \notin S^*$ справедливы равенства $l_{S^*}(x) = f_c(F^*) - f_c(F_-^*(x))$ и $u_{S^*}(y) = f_c(F^*) - f_c(F_+^*(y))$. Таким образом, при известном значении $f_c(F^*)$ для вычисления нижних и верхних допусков вершин x и y достаточно вычислять $f_c(F_-^*(x))$ и $f_c(F_+^*(y))$. Вместе с тем, информацию, накопленную при вычислении числа $f_c(F^*)$, хотелось бы эффективно использовать при нахождении $f_c(F_-^*(x))$ и $f_c(F_+^*(y))$ (а, скажем, не пересчитывать оптимальное независимое множество для графа без/с x/y). Для деревьев такая процедура действительно возможна. В ее основе лежит использование прямого и обратного "хода" динамического программирования.

ЗВНМ для дерева решается рассмотрением этой задачи для поддеревьев с корнями во всевозможных вершинах и сведением задачи для подде-

рева с корнем в вершине x к ЗВНМ для поддеревьев с корнями в сыновьях x. Данный подход был предложен в работе [4]. Для ЗКО $\{\Gamma, c, F, f_c\}$ – ЗВНМ для дерева T с корнем в вершине r рассматриваются всевозможные поддеревья T_x, $x \in V(T)$, где T_x – поддерево T с корнем в вершине x. Далее вводятся следующие переменные: $s(x)$ – оптимальное независимое подмножество поддерева T_x, $s_{\text{in}}(x)$ и $s_{\text{out}}(x)$ – независимые подмножества дерева T_x, имеющие наибольший вес среди всех его независимых множеств, содержащих/не содержащих вершину x соответственно. Множества $s(x)$, $s_{\text{in}}(x)$, $s_{\text{out}}(x)$ имеют веса $w(x)$, $w_{\text{in}}(x)$, $w_{\text{out}}(x)$ соответственно. Заметим, что $w(x) = \max(w_{\text{in}}(x), w_{\text{out}}(x))$ и что $s(x) = s_{\text{in}}(x)$, если $w_{\text{in}}(x) \geq w_{\text{out}}(x)$, и $s(x) = s_{\text{out}}(x)$ в противном случае. Рекуррентные уравнения для весов легко выписываются: если x – лист дерева, то $w_{\text{in}}(x) = c(x)$, $w_{\text{out}}(x) = 0$, а если вершина x не является листом, то $w_{\text{out}}(x) = \sum_{y \in \text{ch}(x)} w(x)$, $w_{\text{in}}(x) = c(x) + \sum_{y \in \text{ch}(x)} w_{\text{out}}(y)$, где $\text{ch}(x)$ – множество непосредственных потомков вершины x во входном дереве. По аналогии выводятся рекуррентные уравнения для множеств $s(x)$, $s_{\text{in}}(x)$, $s_{\text{out}}(x)$.

После получения $w_{\text{in}}(x)$, $w_{\text{out}}(x)$, $w(x)$, $s_{\text{in}}(x)$, $s_{\text{out}}(x)$, $s(x)$ для каждого $x \in V(T)$ (прямым "ходом" динамического программирования) выполняются вычисления всех допусков посредством обратного "хода". Для этого вводятся следующие переменные: $W_{\text{in}}(x) = f_c(F_+^*(x))$, $W_{\text{out}}(x) = f_c(F_-^*(x))$, $S_{\text{in}}(x) \in F_+^*(x)$, $S_{\text{out}}(x) \in F_-^*(x)$. Ясно, что если x – корень T, то $W_{\text{in}}(x) = w_{\text{in}}(r)$, $W_{\text{out}}(x) = w_{\text{out}}(r)$. Заметим, что если $p(x)$ – родитель некорневой вершины x, то $W_{\text{out}}(p(x)) - w(x) = W_{\text{in}}(x) - w_{\text{in}}(x)$ и $W_{\text{out}}(x) - w_{\text{out}}(x) = \max(W_{\text{in}}(p(x)) - w_{\text{out}}(x), W_{\text{out}}(p(x)) - w(x))$. Эти равенства легко переписываются в рекуррентные уравнения. По аналогии можно записать равенства и для $S_{\text{in}}(x)$, $S_{\text{out}}(x)$. Из леммы 1 следует, что для любого $x \in S^* = s(r)$ выполнено равенство $l_{S^*}(x) = w(r) - W_{\text{out}}(x)$ и для любого $y \notin S^*$ выполнено равенство $u_{S^*}(y) = w(r) - W_{\text{in}}(y)$.

Псевдокод решения ЗВНМ для дерева T с корнем r и последующего вычисления всех допусков представлен далее.

ЗВНМ-АЛГОРИТМ + ВЫЧИСЛЕНИЕ ВСЕХ ДОПУСКОВ (T, r)

{
for (в порядке, обратном к обходу T в ширину)
{
if (x является листом)
{
$w_{\text{out}}(x) = 0$; $s_{\text{out}}(x) = \phi$;
$w_{\text{in}}(x) = c(x)$; $s_{\text{in}}(x) = \{x\}$;
$w(x) = c(x)$; $s(x) = \{x\}$;
}
else
{
$w_{\text{in}}(x) = \sum_{y \in \text{ch}(x)} w_{\text{out}}(y) + c(x)$; $s_{\text{in}}(x) = \bigcup_{y \in \text{ch}(x)} s_{\text{out}}(y) \cup \{x\}$;
$w_{\text{out}}(x) = \sum_{y \in \text{ch}(x)} w(y)$; $s_{\text{out}}(x) = \bigcup_{y \in \text{ch}(x)} s(y)$;
$w(x) = \max(w_{\text{in}}(x), w_{\text{out}}(x))$;
if ($w(x) > w_{\text{out}}(x)$) $s(x) = s_{\text{in}}(x)$;
else $s(x) = s_{\text{out}}(x)$;
}
}
for (в порядке обхода дерева T в ширину)
{
if ($x = r$)
{
$W_{\text{out}}(x) = w_{\text{out}}(r)$; $S_{\text{out}}(x) = s_{\text{out}}(r)$;
$W_{\text{in}}(x) = w_{\text{in}}(r)$; $S_{\text{in}}(x) = s_{\text{in}}(r)$;
}
else
{
$W_{\text{in}}(x) = W_{\text{out}}(p(x)) - w(x) + w_{\text{in}}(x)$;
$S_{\text{in}}(x) = S_{\text{out}}(p(x)) \cup (s_{\text{in}}(x) \backslash s(x))$;
$W_{\text{out}}(x) = w_{\text{out}}(x) + \max(W_{\text{in}}(p(x)) - w_{\text{out}}(x), W_{\text{out}}(p(x)) - w(x))$;
if ($W_{\text{out}}(x) - w_{\text{out}}(x) \geq W_{\text{out}}(p(x)) - w(x)$) $S_{\text{out}}(x) = S_{\text{in}}(p(x))$;
else $S_{\text{out}}(x) = S_{\text{out}}(p(x)) \cup (s_{\text{out}}(x) \backslash s(x))$;
}
}
for ($x \in V(T)$)
{
if ($x \in s(r)$) $l_{S^*}(x) = w(r) - W_{\text{out}}(x)$;
else $u_{S^*}(x) = w(r) - W_{\text{in}}(x)$;
}
}

Отметим, что посредством представленного псевдокода вычисление величины $w(r)$ и всех допусков (без соответствующих множеств) осуществляется за время $O(n)$, где n – количество вершин дерева T. Вместе с тем, если множества $s(x)$, $s_{\text{in}}(x)$, $s_{\text{out}}(x)$ хранить одно- или двусвязным списком (это обеспечивает возможность реализации объединения за время $O(1)$), то вычисление $s_{\text{in}}(x)$, $s_{\text{out}}(x)$ осуществляется за время $O(|\text{ch}(x)|)$. Поэтому определение оптимального независимого множества дерева T осуществляется за время $O\left(\sum_{x \in V(T)} |\text{ch}(x)|\right) = O(n)$.

Можно также показать, что общее время вычисления всех множеств $s_{\text{in}}(x) \backslash s(x)$ и $s_{\text{out}}(x) \backslash s(x)$

есть $O(n)$ (для этого необходимо выписать рекуррентные формулы для $s_{\text{out}}(x)\backslash s(x)$ и $s(x)\backslash s_{\text{out}}(x)$, хранить эти множества одно- или двусвязным списками и определять их в первой части псевдокода). Таким образом, трудоемкость всего предложенного псевдокода есть $O(n)$.

Работа выполнена при поддержке Лаборатории алгоритмов и технологий анализа сетевых структур НИУ ВШЭ (грант правительства Российской Федерации, договор 11.G34.31.0057). Первый и второй авторы поддержаны также грантом Научного фонда НИУ ВШЭ (коллективный исследовательский проект “Учитель–Ученики” № 11–04–0008 “Исчисление допусков в задачах комбинаторной оптимизации: теория и алгоритмы”).

СПИСОК ЛИТЕРАТУРЫ

1. *Бойко В., Гольденгорин Б., Кузьменко В.* В кн.: Теорія оптимальних рішень. Киев: Національна Академія Наук України, Ін-т кібернетики імені В.М. Глушкова. 2006. Т. 5. С. 98–104.
2. *Balas E., Saltzman M. J.* // Oper. Res. 1991. V. 39. P. 150–161.
3. *Bekker H., Braad E., Goldengorin B.* // Lect. Notes Comput. Sci. 2005. V. 3503. P. 610–613.
4. *Chen C., Kuo M., Sheu J.* // BIT Numer. Math. 1988. V. 28. P. 353–356.
5. *Germs R., Goldengorin B., Turkensteen M.* // Comput and Oper. Res. 2012. V. 39. № 2. P. 291–298.
6. *Goldengorin B., Jäger G., Molitor P.* // J. Computer Sci. 2006. V. 2. № 9. P. 716–734.
7. *Ramaswamy R., Orlin J., Chakravarti N.* // Math. Program. 2005. V. 102. P. 355–369.
8. *Reinfeld N.V., Vogel W.R.* Mathematical Programming. Prentice-Hall (N.Y.): Englewood Cliffs, 1958.
9. *Shier D. R., Witzgall C.* // Networks. 1980. V. 10. № 4. P. 277–291.
10. *Tarjan R.* // Inform. Process. Lett. 1982. V. 14. P. 30–33.
11. *Turkensteen M., Ghosh D., Goldengorin B., Sierksma G.* // Europ. J. Oper. Res. 2008. V. 189. 3. P. 775–788.
12. *Volgenant A.* // Europ. J. Oper. Res. 2006. V. 169. P. 338–339.