

Результаты ситуационного пересчетного моделирования предложенной системы контроля и управления с использованием натурных данных, полученных в доменном цехе ОАО «ЕВРАЗ Объединенный Западно-Сибирский металлургический комбинат», показали возможность улучшения эффективности управления в 1,5 ÷ 2,0 раза (по интегральному критерию оптимальности) по сравнению с действующей системой.

Литература

1. Андреев, С.М. Оптимизация режима нагрева воздушонагревателей доменных печей / С.М. Андреев, Б.Н. Пар-

сункин // Известия вузов. Черная металлургия. – 2004. – № 7. – С. 33 – 37.

2. Кулаков, С.М. Интеллектуальные системы управления технологическими объектами: теория и практика / С.М. Кулаков, В.Б. Трофимов. – Новокузнецк, 2009.

3. Спирин, Н.А. Информационные системы в металлургии: Учебник для вузов / [Н.А. Спирин и др.]. – Екатеринбург, 2001.

4. Muske, K.R. Blast furnace stove control / [K.R. Muske and others] // American control conference. – Philadelphia: Villanova University, 1998. – С. 24 – 25.

УДК 62

А.Н. Щегряев

Научный руководитель: доктор физико-математических наук, профессор В.В. Захаров

ОБ ОЦЕНКЕ ВРЕМЕННОЙ СОСТОЯТЕЛЬНОСТИ ЭВРИСТИЧЕСКИХ АЛГОРИТМОВ ДЛЯ РЕШЕНИЯ ЗАДАЧ МАРШРУТИЗАЦИИ ТРАНСПОРТА НА БОЛЬШОЙ СЕТИ

В статье рассматриваются различные подходы к оценке эффективности эвристических алгоритмов и приводится новый критерий, названный экспериментальным уровнем временной состоятельности алгоритма.

Временная состоятельность, задача маршрутизации транспорта, эвристические алгоритмы.

Different approaches to assessment of efficiency of heuristic algorithms are considered in the paper. A new criterion called an experimental level of time consistency is presented.

Time consistency, vehicle routing problem, heuristic algorithm.

Введение

Для выживания в условиях свободного рынка компании стремятся использовать весь арсенал доступных инструментов, позволяющих получить конкурентное преимущество перед другими компаниями. Одним из таких инструментов является оптимизация накладных затрат, которая позволяет при том же объеме используемых ресурсов увеличить прибыль. Для транспортных компаний существенная оптимизация затрат может быть достигнута в том числе и за счет построения эффективных маршрутов для транспортных средств. Именно по этой причине эффективным алгоритмам решения задачи маршрутизации транспорта (ЗМТ, VRP – vehicle routing problem) уделено столь пристальное внимание со стороны исследователей. Под задачами транспортной маршрутизации понимают широкий класс известных комбинаторных задач оптимизации. Основной целью в данных задачах является построение набора маршрутов для транспортных средств (ТС), которые обслуживают множество географически распределенных потребителей с заданным спросом. ЗМТ является обобщением широко известной задачи коммивояжера, и поэтому, так же как и задача коммивояжера, относится к классу NP-трудных задач. Таким образом, применение точных алгоритмов для решения задач большой размерности в разумное

время затруднено [6]. Вместо них используют различные эвристические и метаэвристические алгоритмы, способные быстро находить хорошие решения без гарантии оптимальности и, как правило, без оценки погрешности.

Для сравнения эффективности эвристических алгоритмов для определенного класса задач используются опубликованные и находящиеся в открытом доступе тестовые примеры (<http://www.hec.ca/chairedistributique/data>), на которых проверяется работа алгоритма. После этого результаты, полученные при решении тестовых задач определенным алгоритмом, сравниваются с лучшими найденными решениями и оцениваются по нескольким критериям. Алгоритмы постоянно совершенствуются и, таким образом, список лучших известных решений все время обновляется. В данной работе предлагается новый подход к определению эффективности эвристических алгоритмов, основанный на оценке их временной состоятельности.

В части 1 будет произведена краткая постановка рассматриваемой задачи маршрутизации транспорта. После этого в части 2 будет приведено описание уже существующих подходов к оценке эффективности эвристических алгоритмов и предложен новый подход. В части 3 новый подход будет проиллюстрирован на примере алгоритма простого локального по-

иска. В части 4 будут предложены выводы и намечены дальнейшие пути для исследования.

1. Постановка задачи маршрутизации

Впервые задача транспортной маршрутизации была сформулирована в работе Данцига и Рамсера в 1959 г. [4]. Эта задача является ключевой в области перевозок, распределения товаров и логистики. При решении ЗМТ для одного депо, набора потребителей с наперед заданным спросом и множества идентичных транспортных средств (ТС) с заданной грузоподъемностью необходимо построить маршруты движения ТС так, чтобы были выполнены следующие условия:

- каждый маршрут начинается и заканчивается в депо;
- суммарный спрос потребителей, обслуженных одним ТС, не должен превышать грузоподъемность данного ТС;
- все потребители должны быть посещены одним и тем же ТС ровно один раз;
- суммарная длина маршрутов должна быть минимальна.

VRP в классической постановке лишь описывает основную структуру проблемы, но остается малоприменимой для использования в реальных задачах, так как не учитывает очень многих ограничений и условий, которые встречаются в реальной жизни. Именно поэтому в настоящее время пристальному изучению подверглись различные расширения данной задачи, которые лучше описывают реальные транспортные или логистические проблемы. В данной работе мы будем рассматривать ЗМТ с несколькими депо и с временными окнами. Под временными окнами понимается временной интервал, в течение которого должно быть осуществлено обслуживание клиента. У каждого потребителя и у каждого депо есть свое временное окно. Если транспортное средство прибывает раньше начала обслуживания, то оно должно ждать, когда обслуживание станет возможным. Для депо временное окно обозначает период закрытия – время, до которого ТС должно вернуться обратно в свое депо. Для каждого потребителя также задано время обслуживания. Таким образом, на классическую ЗМТ накладываются следующие дополнительные ограничения:

- обслуживание должно быть совершено в определенное временное окно потребителя;
- ТС должно вернуться в депо до его закрытия;
- маршрут начинается и заканчивается всегда в одном и том же депо.

Задача маршрутизации транспорта и ее расширения принадлежат классу NP-трудных задач. Таким образом, нахождение точных решений для задач большой размерности затруднено. Для нахождения хороших решений, близких к оптимальным, используют различные эвристические и метаэвристические алгоритмы. Обзор расширений ЗМТ и соответствующих им алгоритмов решения можно найти, например, в [5].

2. Подходы к оценке алгоритмов

Основным критерием для оценки эффективности алгоритмов является точность. Для определения точности или эффективности эвристического алгоритма используют специальные тестовые примеры, находящиеся в общем доступе. Так как в большинстве случаев оптимальные решения неизвестны, то решения, полученные с помощью исследуемого алгоритма, сравниваются с лучшими известными решениями, полученными с помощью других алгоритмов. К сожалению, не существует единой практики, как происходит само сравнение. Результаты работы многих алгоритмов могут сильно зависеть от выбора параметров алгоритма. Таким образом, один и тот же алгоритм может показывать различные результаты на одном и том же наборе задач с различными параметрами. В то же время при каждом запуске алгоритм может зачастую выдавать разные решения. В результате при публикации результатов оценки алгоритма на эффективность могут быть использованы либо решения, полученные при одном наборе параметров за один запуск алгоритма для всех тестовых задач, либо лучшие решения, полученные за несколько запусков, либо лучшие решения, полученные за несколько запусков при разных параметрах алгоритма для разных тестовых задач. Кроме того, как отмечено в [2], на результаты вычислений могут влиять и различные округления, используемые при вычислениях.

Вторым немаловажным критерием является скорость работы алгоритма. Зачастую скорость работы и точность зависят друг от друга. Многие эвристические алгоритмы являются итерационными, и, таким образом, продолжительность вычислений можно ограничивать максимальным числом итераций или максимальным числом итераций, в течение которых найденное алгоритмом решение не было улучшено. Алгоритмы, работающие быстро, зачастую находят решения хуже, так как менее тщательно исследуют пространство решений и быстро сходятся. Другие алгоритмы тратят больше времени, но при этом находят решения лучше. В реальной практике выбор между быстрыми и медленными алгоритмами осуществляется в зависимости от контекста применения. Например, для маршрутизации карет скоростью помощи медленные алгоритмы совершенно не подходят, но если речь идет о заблаговременном планировании перевозок, то компании могут позволить себе потратить больше времени на разработку маршрутов для того, чтобы сократить транспортные издержки. К сожалению, для многих метаэвристических алгоритмов не применимы результаты теории сложности, поэтому под скоростью работы в данном случае понимают непосредственно вычислительное время. Это время зависит не только от самой схемы алгоритма, но и от таких факторов, как: производительность аппаратных средств, язык программирования, качество программной реализации и структур данных. Таким образом, сравнивать алгоритмы по этому критерию не вполне корректно. Многие исследователи в своих публикациях не приводят время

выполнения алгоритма, лишь говоря о возможных сферах его применения.

Также в работе [2] приводятся еще два дополнительных качественных критерия, важных при оценке и сравнении эвристических алгоритмов: простота, гибкость. В данной работе под простотой автор понимает легкость программной реализации и настройки алгоритма. Чем легче алгоритм для реализации и чем меньше в нем различных параметров, тем удобнее его использовать в реальных приложениях. Именно поэтому такие простые, быстрые, но неэффективные эвристические алгоритмы как алгоритм Кларка-Райта [1] до сих пор используются при решении реальных задач. Под гибкостью автор понимает возможность без серьезных изменений расширить множество ограничений на допустимые решения или расширить модель. Например, насколько легко алгоритм, разработанный для задач маршрутизации транспорта без ограничений по времени, можно адаптировать к задачам с такими ограничениями.

В связи с рассмотренными особенностями предлагается новый критерий для оценки эффективности эвристических алгоритмов для решения задач маршрутизации транспорта, а именно – временная состоятельность.

Для описания процедуры проверки алгоритма на временную состоятельность введем следующие обозначения:

P – множество тестовых задач маршрутизации транспорта;

$p \in P$ – конкретная тестовая задача;

$N(p)$ – множество решений, полученных алгоритмом для задачи p , каждое из которых включает в себя определенный набор маршрутов;

$s(p) \in N(p)$ – решение из множества $N(p)$;

$l(s(p))$ – сумма длин всех маршрутов в решении $s(p)$;

$d(s(p))$ – длительность решения $s(p)$, равная максимальной длительности маршрутов в решении.

Рассмотрим реализацию решения $s(p)$ в динамике. Для этого разобьем интервал времени $[0, d(s(p))]$ на T периодов.

Введем дополнительные обозначения:

$s(t, p)$ – оставшаяся часть решения $s(p)$ задачи p по прошествии t периодов с момента начала выполнения маршрутов. Из данного определения очевидно образом вытекает, что $s(0, p) = s(p)$, а $l(s(T, p)) = 0$, т.е. все ТС после периода T находятся в депо.

$p(s(t, p))$ – подзадача исходной задачи, построенная на основе $s(t, p)$. После того, как решение $s(p)$ будет реализовано в течение t периодов, часть клиентов, обслуженных к этому моменту, можно исключить из начального множества потребителей. При этом транспортные средства, участвующие в реализации решения, будут находиться в соответствующих узлах сети, которые в текущей задаче можно определить как депо. Таким образом, эта подзадача (при участии в реализации решения нескольких транспортных средств) станет задачей маршрутизации с несколькими депо и сокращенным множеством клиентов.

Предположим, что на основе текущего решения $s(t, p)$ сформирована подзадача $p(s(t, p))$. Обозначим эту подзадачу как p' . Получаемое с помощью исследуемого алгоритма решение этой подзадачи обозначим через $s'(p')$.

Определение 1. Если $l(s(p)) \leq l(s'(p'))$, то будем говорить, что решение $s(p)$ сохраняет свойство временной состоятельности на протяжении t периодов в данном эксперименте.

Опишем процедуру оценки временной состоятельности (динамической устойчивости) эвристического алгоритма. Для каждой задачи $p \in P$ проведем следующую процедуру:

1. С помощью исследуемого алгоритма сформируем множество решений $N(p)$.

2. Для каждого решения $s(p) \in N(p)$ определим время $d(s(p))$ и разобьем время реализации на T периодов.

3. Для каждого решения $s(p)$ проведем M следующих экспериментов по проверке свойства временной состоятельности. Будем рассматривать реализацию решения во времени $s(t, p)$, начиная с периода $t = 1$. Сначала сформируем на основе текущего решения $s(t, p)$ подзадачу $p(s(t, p))$. Если $l(s(p)) \leq l(s'(p'))$, то переходим к рассмотрению следующего периода. Если решение $s'(p')$ оказывается лучше исходного подрешения, то мы запоминаем, на каком периоде это произошло и начинаем новый эксперимент с первого периода.

Через $b(s(p), t)$ обозначим функцию, значения которой равны числу экспериментов, в которых решение $s(p)$ потеряло временную состоятельность после реализации периода t . Если бы решение $s(p)$ было оптимальным, в соответствии с принципом оптимальности Беллмана выполнялось условие

$\sum_{t=1}^T b(s(p), t) = 0$. Но эвристические алгоритмы не могут гарантировать оптимальность получаемого решения, поэтому для них мы можем лишь говорить о выполнении следующего неравенства

$$\sum_{t=1}^T b(s(p), t) \leq M.$$

Определение 2. Экспериментальным уровнем временной состоятельности эвристического алгоритма будем называть величину, определяемую следующей формулой:

$$\text{con } L = \frac{\sum_{p \in P} \frac{1}{|N(p)|} \sum_{s(p) \in N(p)} \sum_{t=1}^T b(s(p), t)}{|P| \cdot M}.$$

Исходя из определения, видно, что $0 \leq \text{con } L \leq 1$. Чем ближе это значение к 1, тем меньше решений в ходе экспериментов потеряли свойство временной состоятельности. Таким образом, для алгоритмов, устойчиво генерирующих близкие к оптимальным решения, параметр $\text{con } L$ должен быть близок к 1.

3. Пример

Для иллюстрации применения нового критерия оценки эффективности работы алгоритма с помощью уровня временной состоятельности рассмотрим задачу маршрутизации транспорта с несколькими депо и временными окнами. Для данного класса задач существует опубликованный набор тестовых задач [3]. Всего в этом наборе 20 задач, каждая из которых содержит от 48 до 288 потребителей и от 4 до 6 депо, в которых находится от 1 до 7 транспортных средств. Также задачи отличаются и горизонтами планирования – шириной временных окон. На момент публикации лучшие известные решения можно было найти, например, по ссылке: <http://neo.lcc.uma.es/vrp/vrp-instances/multiple-depot-vrp-with-time-windows-instances/>.

В качестве исследуемого алгоритма рассмотрим простой эвристический алгоритм – локальный поиск, основанный на одном операторе перемещения. Данный алгоритм является итерационным. В ходе каждой итерации алгоритм случайным образом выстраивает очередь из всех потребителей. Каждый потребитель из очереди удаляется из решения. После этого для него ищется такое положение в маршрутах ТС, которое бы увеличивало суммарную длину решения на минимальное число. Другими словами, алгоритм

для каждого потребителя ищет лучшее место в текущем решении. После того, как лучшее положение в решении найдено, потребитель перемещается в него. Данная схема поиска нового решения по своей сути является жадным алгоритмом, и поэтому не претендует на нахождение хороших, близких к оптимальному, решений. Алгоритм прост в реализации и быстро сходится к локальному минимуму. Для нахождения начального решения используется схожая процедура – потребители в случайном порядке добавляются в решение в лучшую позицию.

Результаты экспериментов приведены в таблице. В первом столбце указано название тестового примера. Во втором – средняя длина проверяемых решений из множества $N(p)$. Далее приведены значения $b(s(p), t)$ для девяти периодов. Как было отмечено ранее, проводить эксперименты после реализации последнего периода не имеет смысла, так как к этому моменту все потребители уже будут обслужены. В последнем столбце приведено число решений, сохранивших свойство эффективности. Используя полученные данные, можно вычислить уровень временной состоятельности. В данном иллюстрационном примере $\text{con } L = 0,0662$, что является низким значением.

Таблица

Результаты вычисления для алгоритма локального поиска

Задача	Средняя длина	Периоды									Эффективные решения
		1	2	3	4	5	6	7	8	9	
pr01	1182,7	405	172	126	113	58	3	2	0	0	121
pr02	2183,8	480	137	65	59	56	65	34	0	0	104
pr03	3000,3	571	159	83	71	23	20	26	0	0	47
pr04	3644,0	609	185	86	24	10	13	33	0	0	40
pr05	3952,6	566	191	138	41	18	9	7	24	0	6
pr06	4547,4	552	226	114	21	18	12	11	6	0	40
pr07	1608,9	488	173	87	92	31	59	4	0	0	66
pr08	2646,8	712	153	45	25	7	16	12	13	0	17
pr09	3431,4	537	237	141	36	10	9	7	12	0	11
pr10	4657,8	422	182	141	67	35	12	43	61	0	37
pr11	1178,4	276	151	158	97	58	9	1	0	0	250
pr12	2060,5	488	200	95	46	31	26	20	0	0	94
pr13	2883,1	577	210	44	16	13	37	24	0	0	79
pr14	2999,9	411	197	96	33	21	13	22	1	0	206
pr15	3644,4	569	309	71	21	10	7	7	0	0	6
pr16	4001,4	458	217	120	44	52	12	32	22	0	43
pr17	1646,2	374	185	116	71	69	68	38	0	0	79
pr18	2534,7	458	252	99	40	18	25	43	9	0	56
pr19	3031,3	453	227	110	27	21	59	67	18	0	18
pr20	4705,3	361	449	131	41	8	3	2	1	0	4
Всего		9767	4212	2066	985	567	477	435	167	0	1324

4. Заключение

В ходе исследования был предложен новый критерий оценки качества эвристических алгоритмов для решения задач маршрутизации транспорта – уровень временной состоятельности. Чем больше уровень временной состоятельности, тем выше вероятность того, что найденное с помощью эвристического алгоритма решение не может быть улучшено в ходе реализации с помощью самого алгоритма. Для точных алгоритмов, способных найти оптимальное решение, значение этого критерия будет равно максимальному значению, а именно – 1. Стоит заметить, что уровень временной состоятельности не может быть определен для любых алгоритмов. Рассматриваемый алгоритм должен быть способен решать как саму задачу, так и ее подзадачи, которые в общем случае могут относиться к разным классам задач маршрутизации транспорта. Таким образом, для применения этого критерия исследуемый алгоритм должен обладать определенной гибкостью к накладываемым дополнительным ограничениям. В ходе дальнейших исследований планируется произвести сравнительный анализ различных эвристических и метаэвристических алгоритмов с помощью уровня временной состоятельности.

Литература

1. *Clarke, G.* Scheduling of vehicles from a central depot to a number of delivery points / G. Clarke, J. Wright // *Operations Research*. – 1964. – V. 12. – № 4. – P. 568 – 581.
2. *Cordeau, J-F.* A guide to vehicle routing heuristics / J-F Cordeau, Michel Gendreau, G. Laporte, J-Y. Potvin, F. Semet // *Journal of the Operational Research Society*. – 2002. – V. 53. – P. 512 – 522.
3. *Cordeau, J.F.* A Unified Tabu Search Heuristic for Vehicle Routing Problems with Time Windows / J.F. Cordeau, G. Laporte, A. Mercier // *Journal of the Operational Research Society*. – 2001. – V. 52. – P. 928 – 936.
4. *Dantzig, G. B.* The Truck Dispatching Problem / G.B. Dantzig, R.H. Ramser // *Management Science*. – 1959. – V. 6. – P. 80 – 91.
5. *Gendreau, M.* Metaheuristics for the Vehicle Routing Problem and its Extension: A Categorized Bibliography / Michel Gendreau, Jean-Yves Potvin, Olli Bräysy, Geir Hasle, Arne Løkketangen // *CIRRELT: Document de travail CIRRELT-2007-27*. – 2007.
6. *Lenstra, J.* Complexity of vehicle routing and scheduling problems / J. Lenstra, A. Rinnooy Kan // *Networks*. – 1981. – V. 11. – P. 221 – 228.