ЗАДАЧА РАСПРЕДЕЛЕНИЯ НАГРУЗКИ ПРИ ПАРАЛЛЕЛЬНОЙ РЕАЛИЗАЦИИ РАСШИРЕННЫХ СЕТЕЙ ПЕТРИ В ЗАДАЧЕ МИКРОСКОПИЧЕСКОГО МОДЕЛИРОВАНИЯ ДОРОЖНОГО ДВИЖЕНИЯ

Н.М. Ершов

Московский государственный университет им. М.В.Ломоносова

Введение

Настоящая работа посвящена задаче микроскопического моделирования городского дорожного движения транспорта [1]. Инструментом моделирования были выбраны сети Петри — классическое средство низкоуровневого моделирования распределенных систем [2-4]. Актуальность микроскопического подхода к моделированию дорожного траффика не в последнюю очередь связана с широким развитием параллельных систем программирования. В данной работе рассматриваются вопросы крупноблочной параллельной реализации сетей Петри на многопроцессорных вычислительных системах с использованием технологии MPI. Работа выполнена при финансовой поддержке РФФИ (грант №14-07-00628 А).

Сеть Петри

Предлагается следующая имитационная модель дорожного движения. Предполагается, что схема движения транспорта представляет собой набор линейных участков дорог (с односторонним и однополосным движением), каждый участок характеризуется максимально возможной скоростью движения машин. Точками соединения различных дорог являются узлы-перекрестки (рис. 1).

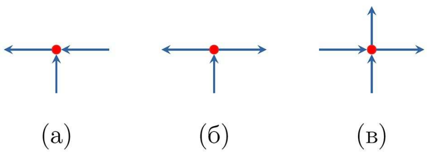


Рис. 1 Примеры некоторых возможных дорожных узлов: а) Т-образный перекресток; б) развилка; в) более сложный узел

Заданная схема движения преобразуется в сеть Петри. Для того каждый линейный участок дороги делится на короткие сегменты (примерно по три метра, предполагается, что на каждом сегменте может располагаться не более одной машины), каждому сегменту ставится в соответствие свое место сети Петри. Машины в такой модели представляются метками, таким образом, любое место содержит не более одной метки, т. е. сеть Петри является безопасной. За логику движения машин отвечают переходы, которые в предлагаемой модели могут быть нескольких типов (рис. 2).

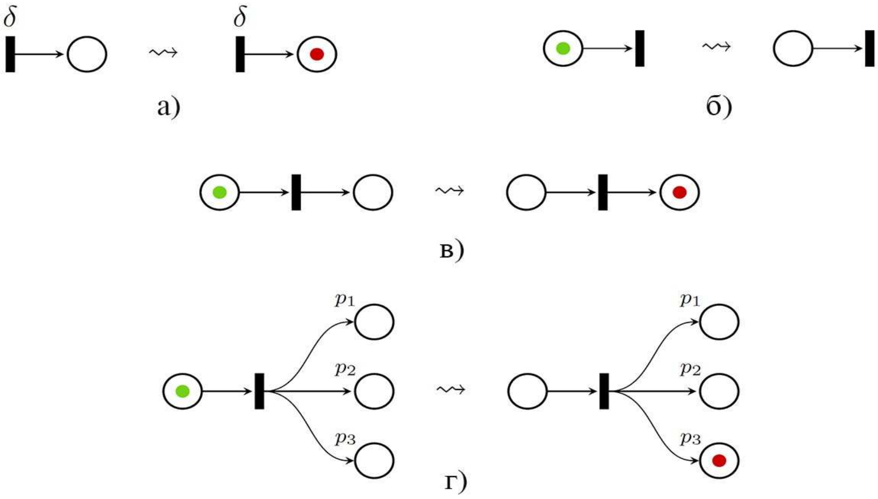


Рис. 2 Различные типы переходов: а) источник; б) сток; в) простой; г) развилка

Например, переходы-источники обеспечивают поступление новых машин в систему, переходы-стоки служат для удаления машин из системы, простые переходы выполняют перемещение машин из входного места в выходное, переходы-развилки перемещают случайным образом машины в одно из своих выходных мест. Для реализации правил типа «уступи дорогу» в модель вводятся специальные блокирующие (ингибиторные) дуги.

После перемещения машины в новое место, на некоторое время (определяемое длиной соответствующего сегмента дороги и скоростью движения машины) машина становится неактивной для последующего переноса. Такой прием позволяет моделировать движение транспорта с заданным скоростным режимом.

Схема распараллеливания

Даже для небольших городов протяженность дорог может составлять сотни километров. Это значит, что соответствующие сети Петри будут содержать десятки и сотни тысяч вершин (мест и переходов). Следовательно, любой вычислительный эксперимент с такой моделью предполагает выполнение очень больших объемов вычислений, что и приводит нас к задаче распараллеливания вычислений.

Рассматривается достаточно очевидная схема распараллеливания, когда вся сеть (по сути, двудольный граф) делится на p фрагментов (по числу имеющихся процессоров системы). Для обеспечения взаимодействия этих фрагментов между собой пограничные места, располагающиеся на стыке двух фрагментов, дублируются, т. е. оказываются принадлежащими сразу двум этим фрагментам (своего рода теневые грани, см. рис. 3).

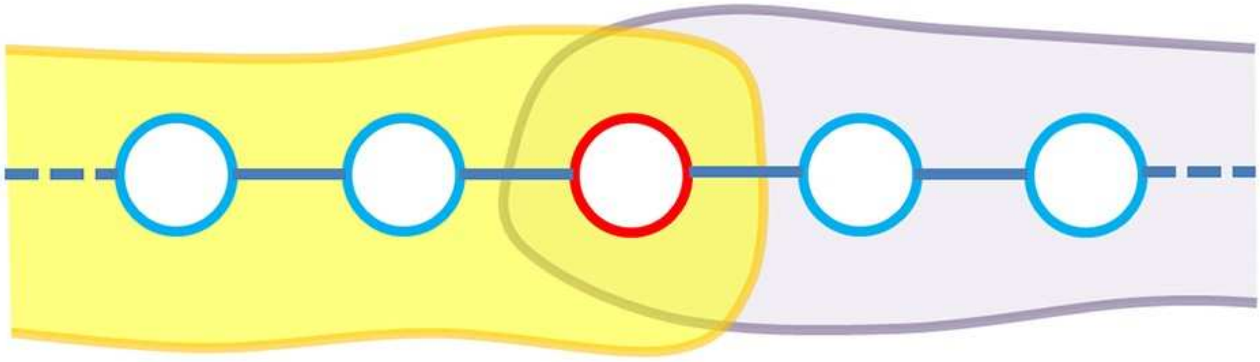
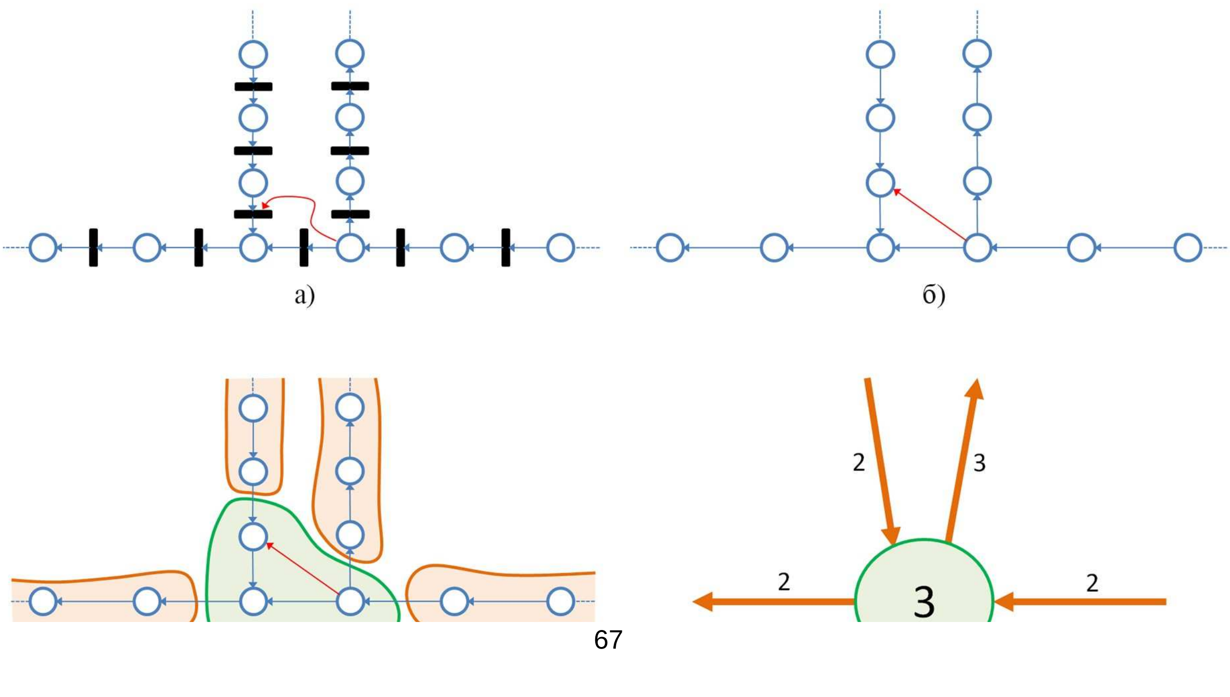


Рис. 3 Схема распределения мест сети Петри

Таким образом, получаем фактически классическую задачу разрезания графов на подграфы примерно одинакового размера с минимальным количеством разрезаемых ребер (теневых мест) [5]. Проблемой, затрудняющей применение стандартных универсальных алгоритмов разрезания, является большой размер возникающих в задаче графов. В настоящей работе предлагается иерархический подход к решению данной задачи, учитывающий ее особенности: граф сети Петри является (почти) планарным, сильно разреженным и содержащим большое число линейных фрагментов.

Алгоритм распределения нагрузки

Предлагаемый алгоритм распределения нагрузки (разрезания сети Петри) состоит из нескольких шагов. На первом шаге исходная сеть Петри (двудольный граф, рис. 4а, красным цветов выделена блокирующая дуга) преобразуется в обычный граф G присоединением всех вершин-переходов ко своим выходным местам (рис. 4б).



На втором шаге строится так называемый макрограф M. Вершинами нового графа являются узлы графа G. В данном контексте, узел — это набор связанных между собой вершин, степень которых не равна 2 (см. рис. 4в). Узлами окажутся, например, источники и стоки (степень 1), а также развилки и слияния (степень больше 2). С точки зрения схемы дорожного движения, узлам соответствуют точки входа/выхода и перекрестки. Вершины графа G степени 2 также объединяются и образуют в результате дуги графа M. Каждая вершина и каждая дуга графа M получает свой вес, равный числу вершин исходного графа G, вошедших в эту вершину или дугу (рис. 4г).

Результатом выполнения описанных двух шагов алгоритма является построение графа, размер которого на несколько порядков меньше размера исходного графа (т. е. размера исходной сети Петри). Для примера рассмотрим граф (рис. 6а), возникающий после применения описанной методики к схеме кольцевого движения (рис. 5). Соответствующая сеть Петри для этой схемы получается кольцевым соединением четырех фрагментов, показанных на рис. 4а.

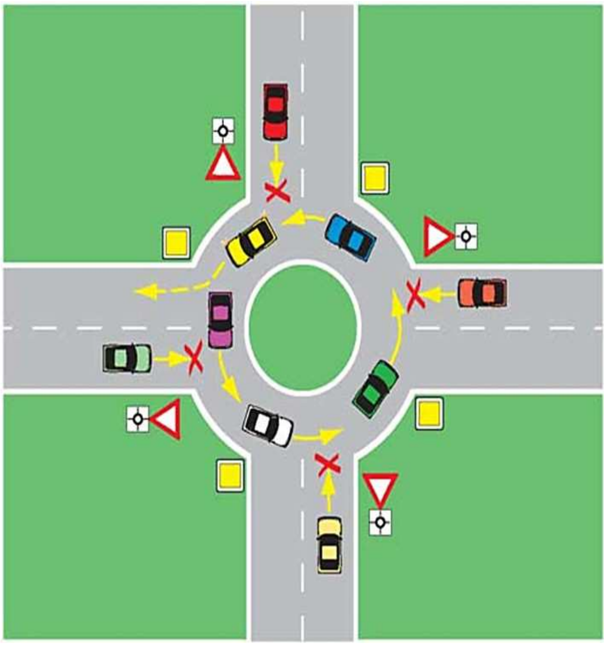


Рис. 5 Пример схемы с круговым движением

Далее рассматривается уже разрезание графа M на p подграфов. Особенностью этой задачи является то, что разрезаемые ребра характеризуются *местом* разрезания, при этом часть ребра до разреза отходит к одной концевой вершине, вторая часть (от разреза) — другой концевой вершине. Задача решается в два этапа.

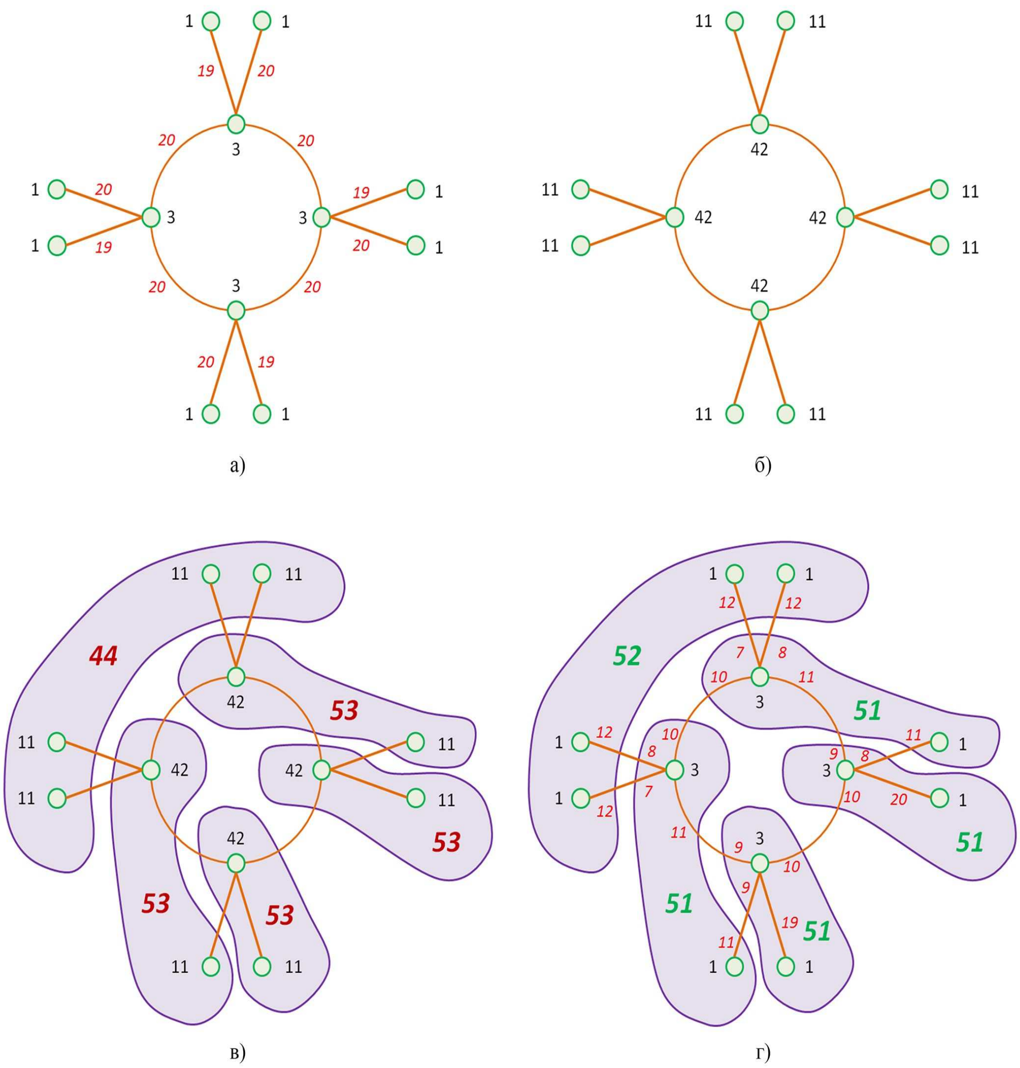


Рис. 6 Пример работы иерархического алгоритма распределения нагрузки

На первом этапе производится распределение вершин по подграфам. Для этого каждое ребро режется ровно пополам, половина веса ребра прибавляется к весу каждой концевой вершины этого ребра (рис. 6б). Получаем новый граф, структура которого совпадает со структурой графа *M*, а взвешенными теперь являются только вершины. Именно для этого графа решается классическая задача разрезания [7]: суммарный вес всех подграфов должен быть примерно одинаковым, а число разрезов — минимальным (рис. 6в).

После выполнения описанного этапа получаем разрез, который однако может быть не оптимальным в смысле его равномерности. Вторым этапом алгоритма производится доводка полученного решения за счет выбора более оптимальных точек разреза ребер (до этого предполагалось, что ребра режутся ровно пополам). Результат работы второго шага показан на рис. 6г.

После того, как разрезание графа *M* построено, оно очевидном образом преобразуется в разрезание исходного графа *G*, т. е. в разрезание исходной сети Петри. Последним шагом является построение расписание выполнения обменов данными между процессорами системы. Определяется в какой последовательности должны выполняться двусторонние обмены данными, и какие именно данные подлежат пересылке. Здесь также возникает (относительно небольшая) оптимизационная задача по минимизации задержек при выполнении обменов данными.

Заключение

Таким образом, исходная большая задача распределения нагрузки для заданной сети Петри оказывается сведенной к трем существенно более простым оптимизационным задачам. Особенностью рассматриваемой задачи (низкоуровневое моделирование дорожного движения) является то, что схема движения (а значит и соответствующая ей сеть Петри) разрабатывается один раз, а затем многократно используется при выполнении численных экспериментов. Следовательно, возникающая при этом задача выравнивания нагрузки оказывается по сути статической, т. е. на ее решение не накладывается жестких временных ограничений. По этой причине, описанный в работе подход был реализован программно на языке Python. При этом, для решения трех промежуточных оптимизационных задач применялись клеточные генетические алгоритмы [8]. Заметим, что описанный в работе иерархический подход к построению сбалансированных распределений нагрузки в принципе может быть применен и к другим графовым моделям, графы которых обладают достаточно четкой макроструктурой.

ЛИТЕРАТУРА:

1. S. P. Hoogendoorn, P. H. L. Bovy "State-of-the-art of vehicular traffic flow modelling" // J. Syst. Cont. Eng., 2001, 215(4), pp. 283-303.
2. Jiacun Wang; Chun Jin; Yi Deng, "Performance analysis of traffic networks based on Stochastic Timed Petri Net models" // Engineering of Complex Computer Systems, Oct. 1999, pp.77-85.
3. Mariagrazia Dotoli, Maria Pia Fanti "An urban traffic network model via coloured timed Petri nets" // Control Engineering Practice, Volume 14, Issue 10, October 2006, Pages 1213-1229.
4. Farhi, Nadir and Goursat, Maurice and Quadrat, Jean-Pierre "Road Traffic Models Using Petri Nets and Minplus Algebra" // Traffic and Granular Flow ’07, 2009, pp 281-286.
5. Shirazi, Behrooz A., Kavi, Krishna M., Hurson, Ali R. "Scheduling and Load Balancing in Parallel and Distributed Systems", IEEE Computer Society Press, 1995.
6. Andreev, Konstantin and Räcke, Harald "Balanced Graph Partitioning" // Proceedings of the sixteenth annual ACM symposium on Parallelism in algorithms and architectures, Barcelona, Spain, 2004: pp. 120-124.
7. Hendrickson, B. and Leland, R. "A multilevel algorithm for partitioning graphs" // Proceedings of the 1995 ACM/IEEE conference on Supercomputing. ACM. p. 28.
8. Ершов Н. "Неоднородные клеточные генетические алгоритмы" // Программные системы и инструменты: тематический сборник, под общей ред. Королева Л.Н. — Т. 13. — Макс-ПРЕСС Москва, 2012. — С. 37–43.