Секция № 2
«Актуальные проблемы фундаментальной и прикладной математики»

Содержание

ДВУМЕРНЫЕ ГЕОМЕТРИЧЕСКИЕ ОБЪЕКТЫ В КОРРЕЛЯЦИОННОМ И ФАКТОРНОМ АНАЛИЗЕ

Козик Е.С., Северюхина Н.А.
Оренбургский государственный университет, г. Оренбург

Основными регулирующими и управляющими факторами, определяющими механические свойства изделий порошковой металлургии (ИПМ) при правильно выбранных режимах с учетом особенностей структурного состояния, деформации, нагрева и охлаждения пористых тел являются: термическая обработка изделий из порошковых сталей.

Особенности структурных превращений при термической обработке оказывает существенное влияние на свойства изделий. Охлаждение эвтектоидной порошковой стали пористостью 14 - 16 % со скоростью 2 - 3 ^{0}С/с приводит к повышению предела прочности при изгибе от 470 до 690 МПа. Увеличение скорости охлаждения до 10 ^{0}С/с способствует измельчению структуры и повышению прочности на изгиб до 750 МПа. Повышение пористости порошковых сталей уменьшает твердость закаленной структуры, а также время устойчивости переохлажденного аустенита. Твердость закаленных с оптимальных температур образцов из порошковых сталей всегда ниже, чем у литых сталей аналогичного состава. Уровень прочностных свойств повышается при легировании никелем и молибденом.

Для решения многокритериальных задач по оптимизации режима термообработки (закалки и отпуска) используются следующие подходы:

1. Ранжировка (упорядочение) факторов, их объединение в группы и сведение решения многокритериальной задачи к решению по одному выбранному фактору (используется преимущественно при линейных зависимостях между переменными).

2. Формирование моделей многокритериальных ситуаций (многомерных образов ситуаций) и ранжирование этих моделей по заданному фактору (используется при невозможности или нецелесообразности приведения взаимных отношений переменных к линейным) [1].

Принята следующую последовательность действий: анализ и разработка структуры системы с оценкой линейности или нелинейности зависимостей варьируемых переменных [2, 3].

Факторный анализ данных о физических, механических и технологических характеристиках образцов из порошковых материалов позволил определить вклад каждого из технологических параметров на качественные характеристики, т.е. вклады предикторов (аргументов) на переменную отклика (зависимую переменную), результаты в таблице 1.

Приведенные результаты отражают воздействие всей системы предикторов на тот или иной отклик. Приведенные оценки позволили выделить указанные предикторы и сосредоточить внимание на этих характеристиках.

Таблица 1 - Сводная таблица оценок степени влияния предикторов на выходные параметры (отклики)

Предикторы		Зависимые переменные (отклики)			
		PORIS	TVERD	UDLIN	BALL
Номер	Название	Вклад	Вклад	Вклад	Вклад
2	SOD_C	**-0.42959**	**0.29121**	**-0.22072**	**0.11725**
3	SOD_CU	-0.00542	-0.03107	-0.03536	**-0.11055**
4	SOD_CR	-0.01280	-0.01209	0.00248	-0.03000
5	SOD_NI	0.00431	0.00501	0.01842	0.00576
7	T_NZ	**0.46322**	**-0.62513**	**-0.66198**	**-0.52222**
8	V_OHL	0.00871	0.02595	0.00519	-0.07152
9	T_OTP	-0.01385	-0.00752	-0.05371	0.06990
13	KOD_MATE	0.06211	-0.00203	0.00213	-0.07281

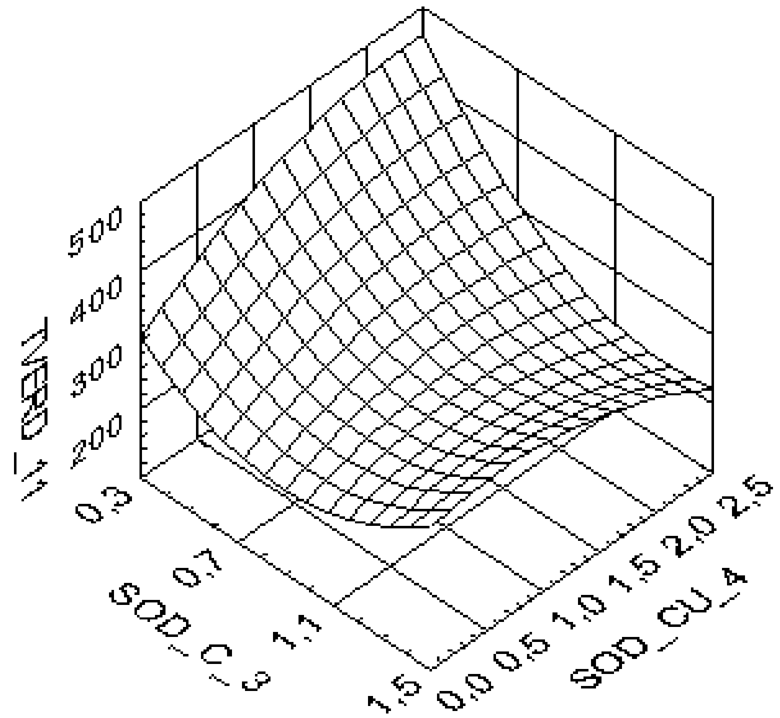
В этих условиях построение моделей и зависимостей, основанных на линейных отношениях, не имеет перспективы, поскольку ошибки будут значительными. Поэтому на данном этапе проводим оценки парных и комбинированных воздействий, позволяющих определить и выделить особенности комплексного взаимодействия переменных, сначала парные, затем более сложные - комбинированные.

Для наглядного представления влияния пар переменных на механические характеристики использовали графики парного влияния процентного содержания компонентов на твердость образцов из порошковых материалов (рисунок 1).

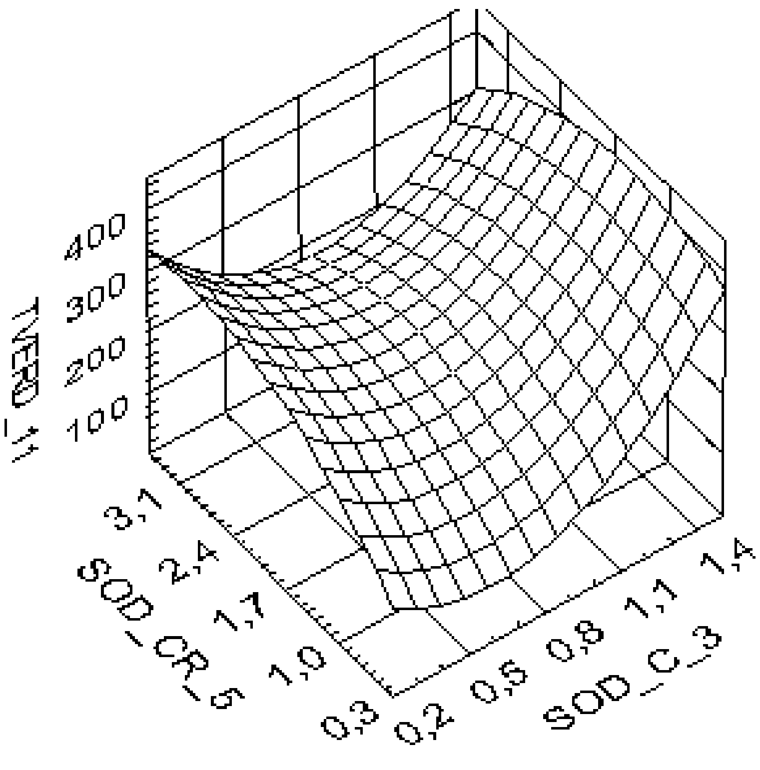
Анализ приведенных графических зависимостей, имеющих седловидный характер, позволяет определить сложное совместное влияние различных составов порошковых материалов и параметров технологических воздействий на контролируемые физические и механические характеристики изделия.

Корреляционный коэффициент содержания углерода и меди равный – 0,47 однонаправлено, воздействуют на твердость образца. При содержании углерода 1,5 % и меди 2,5 % наблюдается максимальная твердость. При содержании углерода 0,3 % увеличение содержания меди снижает твердость образца. При содержании меди 2,5 % уменьшение содержания углерода (с 1,5 % до 0,3 %) приводит к снижению твердости больше чем в 2 раза (480–220 HB).

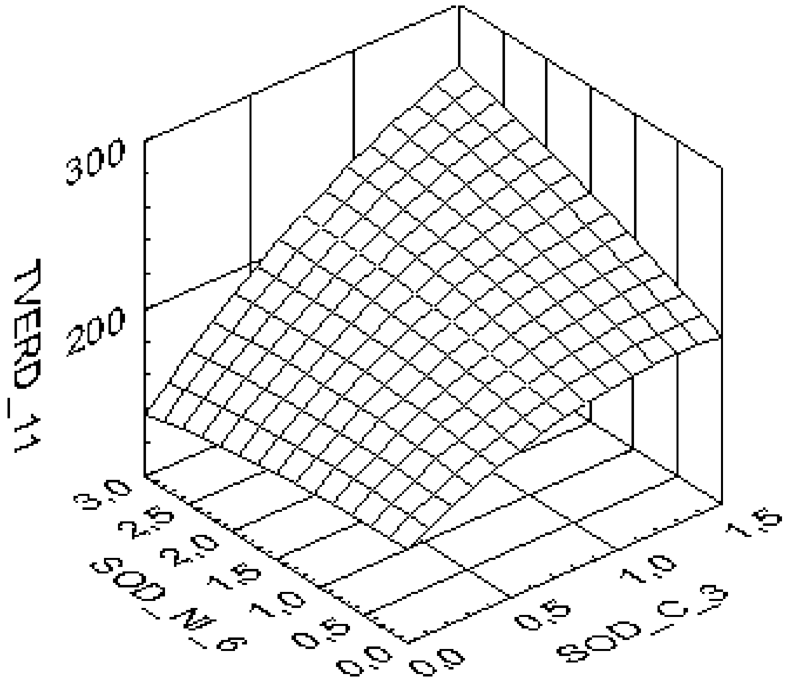
Минимальное значение твердости образцов (200 HB) наблюдается при содержании углерода около 0,7 %, незначительно увеличивается при увеличении содержания меди и достигает 230 HB.


TVERD_11
500
400
300
200
0,3
0,7
1,1
1,5
SOD_C_3
0,0
0,5
1,0
1,5
2,0
2,5
SOD_CU_4

а)


TVERD_11
400
300
200
100
3,1
2,4
1,7
1,0
0,3
SOD_CR_5
0,2
0,5
0,8
1,1
1,4
SOD_C_3

б)



в)

Рисунок 1 - Зависимость твердости (TVERD) от содержания углерода (SOD_C) и меди (SOD_CU) (а), и хрома (SOD_CR)(б) и никеля (SOD_Ni)(в) в %

Максимальное содержание хрома (3,5 %) и минимальное содержание углерода характеризуется твердостью менее 320 НВ, а при увеличении содержания углерода (с 0,2 % до 1,5 %) - 350 НВ. Максимальная твердость соответствует содержанию углерода (1,5 %) и хрома (от 1,0 до 2.4 %). Увеличение содержания хрома от 0,3 до 3,5 % нелинейно увеличивает твердость от 100 до 320 НВ. Заметное снижение твердости происходит при содержании углерода 0,5 – 0,8 % -«седловина». Влияние процентного содержания хрома (от 0,3 до 3,5 %) на твердость незначительно, она практически не изменяется.

Для получения прогнозов на основе проведенного анализа данных используется комплекс поддержки принятия решений, включающий оболочку экспертной системы и базу знаний, обеспечивающий в диалоговом режиме принятие решений на основе концентрации известных, на данный момент времени, знаний.

Список литературы

1. ***Вертинская Н.Д.*** *Математическое моделирование многофакторных и многопараметрических процессов в многокомпонентных системах. – Иркутск: Изд-во ИрГТУ, 2000. - 286 с.*

2. ***Первикова В.Н., Цыпылова Л.А.*** *Графо-аналитический алгоритм поиска оптимума эмпирической многофакторной зависимости при планировании эксперимента первого порядка. - В кн.: Прикладная геометрия и машинная графика в авиастроении -М.: МАИ, 1981, с.11-15.*

3. ***Петров А.В.*** *Моделирование систем: Учеб. пособие. - Иркутск: Изд-во ИрГТУ, 2001. - 267 с.*

О РАДИКАЛЕ ДЖЕКОБСОНА ДЛЯ СПЕЦИАЛЬНЫХ АЛГЕБР ЛИ

Кучеров А. А., Пихтильков С. А., Пихтилькова О. А.
Оренбургский государственный университет, г. Оренбург

В работе дается обзор результатов по радикалу Джекобсона для алгебр Ли.

Радикал Джекобсона $J(R)$ ассоциативного кольца R является одним из основных радикалов в теории колец. Он совпадает с пересечением аннуляторов всех неприводимых модулей или самим кольцом, если неприводимых модулей над ним не существует.

Пусть R – ассоциативное кольцо. Скажем, что идеал $\rho \subseteq R$ является регулярным, если существует $a \in R$ такое, что $x - ax \in \rho$ для всех $x \in R$.

Через регулярные идеалы можно дать эквивалентное определение: радикал Джекобсона совпадает с пересечением всех максимальных регулярных правых идеалов кольца R[1].

Для колец с единицей любой идеал (правый, левый, двусторонний) является регулярным. Следовательно, в кольце с единицей радикал Джекобсона совпадает с пересечением максимальных правых идеалов.

В алгебрах Ли, в силу антикоммутативности, любой идеал является правым, левым и двусторонним. По аналогии с ассоциативными кольцами с единицей, было выбрано следующее определение радикала Джекобсона для алгебр Ли: назовем радикалом Джекобсона $J(L)$ алгебры Ли L пересечение всех максимальных идеалов.

Отметим, что для конечномерной алгебры Ли над полем характеристики нуль нильпотентный радикал совпадает с радикалом Джекобсона [2].

Для бесконечномерных алгебр Ли свойства радикала Джекобсона исследовали Kamiya, Noriaki [3] и F. Kubo [4].

В частности, было доказано, что если $\sigma(L)$ – наибольший локально разрешимый идеал алгебры Ли L над полем F характеристики нуль, порожденной конечномерными идеалами, то $[L, \sigma(L)] = J(L)$.

В общем случае этот результат неверен даже для локально конечных алгебр Ли [4].

Отметим, что для произвольной алгебры Ли L наибольший локально разрешимый идеал может не существовать [5].

F. Kubo также изучал вопрос существования разложения Леви-Мальцева для локально конечных алгебр Ли с нулевым радикалом Джекобсона.

В 1963 г. В. Н. Латышев ввел новый класс алгебр Ли [6], которые он назвал специальными по аналогии с йордановыми алгебрами.

Скажем, что алгебра Ли L специальная или *SPI*-алгебра Ли, если существует ассоциативная *PI*-алгебра A такая, что L вложена в $A^{(-)}$ как алгебра Ли, где $A^{(-)}$ – алгебра Ли, заданная на A с помощью операции коммутирования $[x,y]=xy-yx$.

Нам потребуется понятие присоединенной ассоциативной алгебры.

Пусть L – алгебра Ли, $a \in L$. Через *ad a* обозначим линейное отображение $ad\,a : L \to L$, заданное формулой *x ad a=[x,a]*. Обозначим через *Ad L* ассоциативную алгебру, порожденную в *End L* множеством $\{ad\,a |\ \ a \in L\}$ и назовем ее присоединенной ассоциативной алгеброй к алгебре Ли L.

Скажем, что алгебра Ли является обобщенно специальной, если *Ad L* является *PI*-алгеброй. Для каждой специальной алгебры Ли L присоединенная алгебра *Ad L* является *PI*-алгеброй [7].

Обобщенно специальной является любая алгебра Ли, удовлетворяющая тождествам специальной алгебры Ли [8]. Согласно теореме Адо, любая конечномерная алгебра Ли является специальной [9].

Следовательно, обобщенно специальной является любая алгебра Ли удовлетворяющая тождествам конечномерных алгебр Ли. Это достаточно широкий класс алгебр, в том числе и конечномерных.

Специальные алгебры и обобщенно специальные алгебры Ли являются обобщениями конечномерных алгебр, которые наследуют многие их свойства. К их числу относится результат Ю.А. Бахтурина о подалгебрах Леви.

Алгебра Ли L называется почти разрешимой, если в ней существует разрешимый идеал R такой, что алгебра L/R — конечномерная. Соответственно назовем алгебру Ли почти локально разрешимой, если она является конечным расширением локально разрешимой алгебры Ли.

Пусть R – разрешимый радикал алгебры Ли L. Назовём полупростую конечномерную подалгебру G алгебры L подалгеброй Леви, если G дополняет до L радикал $R(L)$.

Ю.А. Бахтурин доказал для специальных алгебр Ли аналог классической теоремы Леви-Мальцева, справедливой для конечномерных алгебр Ли над полем характеристики нуль.

Теорема 1. (Ю.А. Бахтурина, [8]). Пусть L – конечно порождённая почти разрешимая специальная алгебра Ли над полем характеристики нуль. Тогда в L существует подалгебра Леви.

Результат Ю.А. Бахтурина был обобщен Ю.А. Тереховой.

Теорема 2. (Ю.А. Тереховой, [10]). Пусть L – почти локально разрешимая специальная алгебра Ли над полем характеристики нуль. Тогда в L существует подалгебра Леви.

В процессе доказательства теоремы Ю.А. Тереховой была использована теорема 1 и результат Ю.А. Бахтурина о *PI*-оболочках конечномерных алгебр Ли [11].

Пусть M некоторый L-модуль. Обозначим через $A(M)$ ассоциативную алгебру, порожденную элементами алгебры L в алгебре *End M*.

Назовем *PI*-представлением алгебры Ли L представление алгебры L в алгебре эндоморфизмов $End\,M^{(-)}$ модуля M над алгеброй L, для которого ассоциированная алгебра представления $A(L)$ является PI-алгеброй.

Для специальных алгебр Ли один из авторов ввел понятие локально нильпотентного радикала, обобщающего понятие нильпотентного радикала для конечномерных алгебр Ли [12].

Назовем локально нильпотентным радикалом *N(L)* специальной алгебры Ли *L* пересечение наибольших идеалов локальной нильпотентности всех *PI*-представлений алгебры *L*. Оказалось, что для специальной алгебры Ли *L* радикал *N(L)* наследует некоторые свойства нильпотентного радикала для конечномерных алгебр Ли. В частности, *N(L)* в этом случае является локально нильпотентным.

Известно, что существует специальная алгебра Ли, локально нильпотентный радикал которой строго содержится в радикале Джекобсона [12].

На Международной алгебраической конференции, посвященной 100-летию со дня рождения А. Г.Куроша (Москва, МГУ им. М. В. Ломоносова, 2008) Борис Исаакович Плоткин поставил вопрос о гомологическом описании радикала Джекобсона для алгебр Ли.

Вопрос о возможности гомологического описания радикала Джекобсона был исследован в работе [14].

Обозначим через *Irr(L)* пересечение аннуляторов неприводимых модулей над алгеброй Ли *L* и саму алгебру *L* если их нет.

Обозначим через *IrrPI(L)* пересечение аннуляторов всех неприводимых *PI*-представлений алгебры Ли *L* и саму алгебру *L* если их нет.

Ю. А. Бахтурин использовал пересечение аннуляторов конечномерных неприводимых представлений для доказательства теоремы 1 [8]. Известно, что для конечномерных алгебр Ли пересечение аннуляторов конечномерных представлений совпадает с нильпотентным радикалом [9] и, следовательно, с радикалом Джекобсона для поля нулевой характеристики [2].

Обозначим через *IrrFin(L)* пересечение аннуляторов всех неприводимых конечномерных модулей над алгеброй Ли *L* и саму алгебру *L* если их нет.

Справедлив следующий результат о радикале Джекобсона для алгебр Ли [13].

Теорема 3. Для произвольной специальной алгебры Ли *L* над полем *F* характеристики нуль имеет место включение $J(L) \supseteq IrrPI(L)$.

В работе [14] доказаны следующие утверждения.

Теорема 4. Пусть *L* – произвольная алгебра Ли. Тогда

$$Irr(L) = L \cap J(U(L)),$$

где *U(L)* – универсальная обертывающая алгебра.

Теорема 5. Пусть *L* – конечномерная алгебра Ли над полем *F* характеристики нуль. Тогда *IrrPI(L)=N(L)*, где *N(L)* – нильпотентный радикал алгебры Ли *L*.

Теорема 6. Пусть основное поле имеет нулевую характеристику. Тогда справедливы следующие включения, которые в общем случае строгие:

$$Irr(L) \subset IrrPI(L) \subset IrrFin(L).$$

В работе [14] было предложено считать радикалом Джекобсона алгебры Ли *L* пересечение аннуляторов всех неприводимых *PI*-представлений алгебры Ли *L* и саму алгебру *L* если их нет. Из теоремы 5 и результата Маршалла следует, что *IrrPI(L)=J(L)* для конечномерных алгебр Ли над полем характеристики нуль.

В заключение сформулируем следующие вопросы.

1. Как соотносятся между собой *N(L)* и *IrrPI(L)*?

2. Как соотносятся между собой *IrrPI(L)* и первичный радикал *P(L)*?

3. Обобщим определение радикала *N(L)* на произвольные алгебры Ли *L*. Будет ли идеал *N(L)* локально нильпотентным?

Список литературы

1. ***Херстейн И.*** *Некоммутативные кольца.- М.: Мир, 1972.- 192 с.*
2. ***Marshall E. I.*** *The Frattini subalgebras of a Lie algebra /Marshall E. I.// J. London Math. Soc.- 1967.- V. 42.- P. 416-422.*
3. *Kamiya On the Jacobson radicals of infinite dimensional Lie algebras /Kamiya, Noriaki// Hiroshima Math. J.- 1979.- V. 9.- P. 37-40.*
4. ***Kubo F.*** *Infinite-dimensional Lie algebras with null Jacobson radical /Kubo F.// Bull. Kyushu Inst. Technol. Math. Nat. Sci.- 1991.- V. 38.- P. 23-30.*
5. ***Латышев В.Н.*** *О сумме локально разрешимых идеалов алгебр Ли /Латышев В.Н., Михалев А.В., Пихтильков С.А.// Вестник МГУ.- Сер. 1. матем., мех.- 2003.- 3.- С. 29-32.*
6. ***Латышев В. Н.*** *Об алгебрах Ли с тождественными соотношениями /Латышев В. Н.// Сиб. мат. журнал.- 1963.- Т. 4.- 4.- С. 821-829.*
7. ***Пихтильков С.А.*** *О специальных алгебрах Ли /Пихтильков С.А.// Успехи матем. наук.- 1981.- Т. 36.- 6.- С. 225-226.*
8. ***Бахтурин Ю. А.*** *Тождества в алгебрах Ли.- М.: Наука, 1985, 448 с.*
9. ***Бурбаки Н.*** *Группы и алгебры Ли (главы I-III).- М.: Мир, 1976.- 496 с.*
10. ***Терехова Ю.А.*** *О теореме Леви для специальных алгебр Ли /Терехова Ю.А.// Алгоритмические проблемы теории групп и полугрупп.- Тула: ТГПУ, 1994.- С. 97-103.*
11. ***Бахтурин Ю.А.*** *О строении PI-оболочки конечномерной алгебры Ли /Бахтурин Ю.А.// Изв. вузов. Матем.- 1985.-* $11. - C. 60 - 62.$
12. ***Пихтильков С. А.*** *О локально нильпотентном радикале специальных алгебр Ли /Пихтильков С. А.// Фундаментальная и прикладная математика.- 2002.- Т. 8.- Вып. 3.- С. 769-782.*
13. ***Пихтильков С. А.*** *О некоторых классических радикалах для специальных алгебр Ли /Пихтильков С. А., Пихтилькова О. А.// Чебышевский сборник.- 2008.- Т. 9.- Вып. 1(25).- С. 120-124.*
14. ***Кучеров А. А.*** *О гомологическом описании радикала Джекобсона для алгебр Ли /Кучеров А. А., Пихтильков С. А., Пихтилькова О. А.// Чебышевский сборник (сдано в печать).*

СОВРЕМЕННЫЕ АЛГОРИТМЫ ПЛАНИРОВАНИЯ ЗАДАЧ НА КЛАСТЕРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ

Полежаев П.Н.
Оренбургский государственный университет, г. Оренбург

В настоящее время кластерные вычислительные системы стали наиболее распространенной аппаратной основой для проведения сложных вычислительных экспериментов, создания виртуальных испытательных стендов, а также решения ресурсоемких вычислительных задач, позволяющих их эффективное распараллеливание. Так, в списке Top500 самых производительных параллельных вычислительных систем мира превалирующее большинство составляют кластеры, в рамках проекта «СКИФ-ГРИД» было построено больше десятка современных мощных кластерных систем, установленных в университетах различных городов России.

Однако, даже не очень производительные кластерные системы весьма дороги, и, чтобы окупить затраты на их покупку, необходимо максимально рационально использовать доступные вычислительные мощности. Это может быть сделано за счет применения эффективных алгоритмов планирования параллельных задач, поступающих в очередь управляющей системы вычислительного кластера (УСВК).

Каждая параллельная задача имеет следующие параметры: количество необходимых для ее исполнения вычислительных процессоров, оценку времени ее выполнения, а также, при необходимости, требования к оперативной и дисковой памяти вычислительных узлов. Планировщик вычислительного кластера, в соответствии с заложенным в него алгоритмом планирования, на основе сведений о хранящихся в очереди задачах, должен составлять расписание их запуска, т.е. определять для каждой задачи вычислительные узлы и момент времени ее запуска.

Большинство задач планирования являются NP-полными [1]. Это означает, что составление оптимальных по заданному критерию (или критериям в случае многокритериальной оптимизации) расписаний требует экспоненциального времени, что недопустимо для планировщика УСВК, т.к. каждый цикл планирования будет занимать значительный промежуток времени. Данную проблему решают эвристические алгоритмы планирования, строящие субоптимальные расписания за приемлемое полиномиальное время. Среди них списочные и метаэвристические алгоритмы планирования.

Списочные алгоритмы планирования упорядочивают поступающие в очередь кластера параллельные задачи по некоторому их параметру, например, по количеству необходимых для исполнения процессоров или времени ее выполнения. А затем просматривают ее, начиная в головы, и назначают задачи на свободные вычислительные ресурсы.

Наиболее известные списочные алгоритмы планирования [2-4]: *First Come First Served (*первым пришел, первым обслужен), *First Come First Served*

Scan (первым пришел, первым обслужен со сканированием очереди), *Shortest Job First* (кратчайшая задача первая), *Shortest Job First Scan* (кратчайшая задача первая со сканированием), *Longest Job First* (длиннейшая задача первая), *Longest Job First Scan* (длиннейшая задача первая со сканированием), *Most Processors First Served* (задача с наибольшим количеством процессоров первая), *Most Processors First Served Scan* (задача с наибольшим количеством процессоров первая со сканированием), *Least Processors First Served* (задача с наименьшим количеством процессоров первая), *Random Job First* (случайная задача обслуживается первой) и др.

Нередко вместе с описанными выше списочными алгоритмами планирования рассматривают различные **методы назначения задач на узлы**, которые для выбранных задач определяют по некоторому принципу наиболее подходящие для исполнения свободные вычислительные узлы. Наиболее известные методы назначения задач на вычислительные узлы [5]: *First Fit* (первый подходящий), *Best Fit* (наилучший подходящий), *Fastest Node First* (самый быстрый узел первый), *Least Utilized Node First* (наименее загруженный узел первый), *Random First* (случайный узел первый).

Если планирование вычислительных задач осуществляется на кластере рабочих станций, то нередко совместно со списочными алгоритмами планирования или алгоритмом планирования группы задач (gang scheduling) [3] используется **критерий доступности узлов для планирования** – предикат, использующий для своего вычисления статические и динамические параметры узла кластера и определяющий возможность его использования для исполнения задачи. Как правило, проверяется, чтобы свертка загруженности оперативной памяти, дисковой памяти и процессора не превышала некоторого порогового значения.

Большинство списочных алгоритмов при составлении расписаний оставляют большое количество свободных окон, что приводит к простаиванию вычислительных узлов, а, следовательно, к низкой средней вычислительной загруженности кластера.

Для решения этих проблем в Аргонской национальной лаборатории был предложен **агрессивный вариант алгоритма Backfill (алгоритма обратного заполнения)** [5, 6]. Он преследует две конфликтующие цели – повышение эффективности использования вычислительных ресурсов путем заполнения пустых окон в расписании и предотвращение «подвисания» задач за счет механизма резервирования. Ожидающие задачи хранятся в очереди и упорядочены согласно приоритетам. В каждом цикле планирования ресурсы выделяются задачам в порядке их приоритетов, причем задача может получить вычислительные ресурсы, если они не были отведены k самым приоритетным заданиям.

Д. Фейтельсон и А. Вейл из Еврейского университета г. Иерусалима предложили **консервативный вариант алгоритма Backfill** [6], который отличается от агрессивного тем, что задача не может задерживать запуск всех более приоритетных задач, т.е. она может получить вычислительные ресурсы только, если они не были отведены более приоритетным заданиям.

Исследователи показали, что консервативный вариант алгоритма Backfill по загруженности вычислительных узлов не уступает агрессивному варианту, и, в отличие от последнего, позволяет делать точные предположения о времени запуска каждой задачи, находящейся в очереди, что, в свою очередь, дает возможность точного резервирования ресурсов для запуска задачи сразу же после ее поступления в очередь. Последнее особенно актуально, если кластер включен в состав грид-системы [2, 3, 6].

Различными исследователями было предложено множество модификаций консервативного алгоритма Backfill. Так Б. Лоусон и Е. Смирни предложили вариант для нескольких очередей с приоритетами [7], Д. Перкович и П. Келехер рассматривали рандомизацию положения задач в очереди [8].

Для работы алгоритма Backfill существенным является наличие оценок времени выполнения задач, поступающих в очередь [6]. При большой неточности в оценках падает эффективность работы алгоритма – снижается средняя загруженность вычислительных узлов и увеличивается значение среднего ограниченного замедления. В связи с этим актуальна проблема получения достаточно точных оценок времени выполнения задач.

Основные **способы получения оценок времени исполнения задач**: профилирующие прогонки с последующей экстраполяцией результатов [2], использование нейронных сетей для прогнозирования времени исполнения задачи на основе истории предыдущих запусков, использование оценок, выполняемых пользователем [6].

Современные управляющие системы вычислительного кластера и внешние планировщики для них применяют в основном различные модификации алгоритма Backfill.

К числу **метаэвристических алгоритмов планирования**, адаптируемых для составления расписания выполнения задач на кластерной системе, относятся алгоритмы глобального поиска, решающие задачу дискретной комбинаторной оптимизации, в том числе алгоритм имитации отжига, алгоритм табу-поиска, генетические алгоритмы и др.

Обозначим за $C(S)$ целевую оптимизируемую функцию, определяющую качество расписания S. Чаще всего в качестве функции $C(S)$ используют время завершения последней исполняющейся задачи.

*Алгоритм имитации отжига***.** Данный алгоритм строит последовательность расписаний $S_0, S_1, ..., S_k$. Начальное расписание S_0 выбирается некоторым случайным образом. Переход от расписания S_i к расписанию S_{i+1} ($i = \overline{0, k-1}$) осуществляется следующим образом: к расписанию S_i случайно применяется одна из операций модификации (например, смена вычислительного процессора, выделенного некоторой задаче или изменение порядка запуска вычислительных задач) в результате чего получается промежуточное расписание S', которое принимается в качестве расписания S_{i+1} с вероятностью $p(S', S_i)$, определяемой по формуле:

$$p(S', S_i) = \begin{cases} 1, & C(S') - C(S_i) \le 0; \\ e^{-\frac{C(S') - C(S_i)}{q_i}}, & C(S') - C(S_i) > 0, \end{cases}$$

где $\{q_i\}_{i=\overline{0,\infty}}$ - некоторая убывающая положительная последовательность.

На каждом шаге алгоритм сравнивает текущее расписание S_i с наилучшим найденным S^* и обновляет его, если текущее расписание лучше. Возможны разные варианты остановки исполнения алгоритма, например, достижение заданного числа итераций или окончание отведенного времени выполнения.

Одним из недостатков данного алгоритма является то, что возможен возврат к расписаниям, рассмотренным ранее. Это может привести к колебаниям вокруг локального минимума, в результате будет рассмотрена лишь небольшая часть пространства расписаний.

Алгоритм имитации отжига применяется для построения многопроцессорных расписаний в работе [9].

Алгоритм табу-поиска. Данный алгоритм также как и предыдущий строит последовательность расписаний $S_0, S_1, ..., S_k$. Он также строит и постоянно обновляет список Γ некоторых атрибутов рассмотренных ранее расписаний. Переход от расписания S_i к расписанию S_{i+1} осуществляется следующим образом: к расписанию S_i применяются все возможные операции модификации из заданного набора – в результате строится множество Θ расписаний-результатов; затем из этого множества исключаются все расписания, которые характеризуются атрибутами из Γ, в результате получается множество Θ'; после чего из Θ' случайно выбирается расписание S_{i+1}.

Использование списка атрибутов рассмотренных ранее расписаний позволяет избегать возврата к исследованным расписаниям, тем самым увеличивая исследуемую алгоритмом часть комбинаторного пространства, а, следовательно, также увеличивая шансы нахождения глобального оптимума.

В работе [1] рассматривается модификация данного алгоритма, когда с каждым атрибутом списка Γ дополнительно связывается критерий стремления, позволяющий нарушать соответствующий табу-атрибут.

Генетический алгоритм. Каждое расписание S представляется в виде хромосомы (особи) $S = (s_1, s_2, ..., s_m)$, гены s_i которой отражают некоторые составляющие элементы расписания, в зависимости от используемых моделей задачи и вычислительной системы. Например, в работе [10] ген s_i хранит номер процессора вычислительной системы, на который отображается i-ая задача, в источнике [4] ген включает в себя пару значений (p_i, o_i), привязанную к i-й задаче, где p_i - номер вычислительного узла, на котором будет исполняться i-ая задача, o_i - порядковый номер задачи на p_i-м вычислительном узле. Для удобства выполнения операций скрещивания и мутации иногда хромосомы вместе со своими генами кодируются в виде последовательностей бит фиксированной или переменной длины.

На первом шаге алгоритм строит случайно некоторую начальную популяцию хромосом-расписаний $P_0 = (S_{01}, S_{02}, ..., S_{0q})$. В ходе работы алгоритма создается последовательность популяций $P_0, P_1, ..., P_k$, переход от популяции P_i к P_{i+1} осуществляется следующим образом:

1. Для каждой хромосомы популяции P_i вычисляется ее приспособленность – обратное значение целевой функции $\frac{1}{C(S)}$.

2. Пропорционально приспособленностям из популяции P_i несколько раз случайно выбираются пары расписаний и между ними осуществляется операция скрещивания, заключающаяся в случайном обмене генами между выбранными хромосомами. К получившимся потомкам с заданной вероятностью применяется операция мутации, заключающаяся в случайной небольшой модификации генетического материала. В результате применения данных операций создается промежуточная популяция P', содержащая хромосомы популяции P_i и их потомков.

3. Для формирования популяции P_{i+1} осуществляется отбор (селекция) хромосом из промежуточной популяции P', который производится таким образом, чтобы из P' преимущественно удалялись наименее приспособленные особи.

Работа алгоритма обычно останавливается при достижении максимального количества итераций или в случае незначительного изменения средней приспособленности популяции на протяжении нескольких поколений.

Различные варианты сочетаний генетического алгоритма планирования с алгоритмом имитации отжига и алгоритмом табу-поиска, осуществляющие планирование задач в грид-системе, описаны в работе [11]. Использование генетических алгоритмов для построения расписаний, учитывающих сетевые задержки, отражено в [12].

Алгоритмы планирования, учитывающие внутреннюю структуру задач. Описанные ранее эвристические алгоритмы могут использоваться не только для планирования на уровне ресурсов, но и на уровне приложения, когда известна внутренняя структура задачи, представленная, например, с мощью ациклического орграфа подзадач [13]. Списочные алгоритмы могут быть расширены за счет дополнительного учета ограничений предшествования на множестве подзадач, или с помощью задания дополнительного алгоритма, определяющего, например, с помощью топологической сортировки вершин орграфа подзадач, порядок их добавления в очередь.

Метаэвристические алгоритмы планирования при составлении начального расписания и выполнения операций модификации расписаний могут также, как и списочные алгоритмы, дополнительно учитывать ограничения предшествования. Так в работе [10] для построения отображения подзадач на вычислительные узлы используется генетический алгоритм, который для вычисления целевой функции на каждой итерации по хромосоме, задающей отображение подзадач на вычислительные процессоры, целиком строит расписание, последовательно вычисляя времена запуска подзадач.

Отдельным классом можно выделить *алгоритмы планирования, рассматривающие критический путь орграфа подзадач* [4, 13]. Как правило, основная их идея заключается в планировании в первую очередь вершин и дуг, входящих в состав наиболее длинного пути орграфа подзадач.

Алгоритмы, учитывающие отношения предшествования между задачами. Алгоритмы планирования, опирающиеся на модель, учитывающую отношения предшествования между задачами, как правило, аналогичны алгоритмам, учитывающим представление задачи в виде ориентированного ациклического графа подзадач, с той лишь разницей, что вершинами орграфа являются сами задачи, и задачи поступают в УСВК по ходу работы вычислительной системы (онлайновые алгоритмы планирования) [1, 2, 13].

Алгоритмы планирования, учитывающие сеть вычислительной системы. Топологию вычислительной системы и сетевые издержки учитывают немногие алгоритмы планирования. Даже из современных УСВК, только LSF HPC учитывает топологию сети кластера, пытаясь назначить параллельную задачу на группу максимально топологически близких узлов. Возможные сетевые издержки во всех УСВК учитываются неявно, временные затраты на них включаются в оценки времени выполнения работ.

В работе [10] рассматривается модифицированный генетический алгоритм составления расписаний для вычислительной системы заданной в виде неориентированного взвешенного графа. На каждой итерации алгоритма для вычисления целевой функции, представляющей собой время завершения последней подзадачи, строится расписание запуска подзадач, учитывающее ограничения предшествования, а также коммуникационные задержки при пересылке данных.

Похожий алгоритм описывается в работе [13], но в отличие от предыдущего данный алгоритм, во-первых, рассматривает более сложную модель вычислительной системы, представленную в виде взвешенного ориентированного графа с гиперребрами. Во-вторых, также, как вершины орграфа задачи планируются на вычислительные процессоры, дуги орграфа планируются на простые ориентированные цепи связей между соответствующими вычислительными узлами в рамках заданного графа топологии вычислительной системы.

Также в литературе можно найти узкоспециализированные алгоритмы планирования для конкретных топологий вычислительной системы [1]: в виде решетки, гиперкуба, тора.

Все рассмотренные ранее алгоритмы относятся к **классу алгоритмов планирования с разделением пространства вычислительных ресурсов.** Другой класс формируют **алгоритмы разделения времени**, позволяющие нескольким выполняющимся задачам совместно использовать ресурсы общих вычислительных узлов.

К классу алгоритмов разделения времени относится *алгоритм планирования групп задач (Gang scheduling)* [3, 4], осуществляющий распределение ресурсов вычислительной системы между группами задач. Задачи объединяются в группы, так, что все задачи одной группы одновременно выполняются на вычислительных узлах. Алгоритм осуществляет попеременно переключение групп задач, выделяя каждой группе некоторый квант времени. В результате чего текущие выполняемые задачи приостанавливаются (при небольшом кванте) или даже сбрасывают в своп вычислительных узлов (при значительной величине кванта).

Данный алгоритм применяется, прежде всего, в многопроцессорных вычислительных системах, его реализация для кластера имеет определенные трудности с синхронизацией переключения задач одновременно на всех узлах, в частности, возможно, потребуется модификация планировщика процессов операционной системы вычислительных узлов. Он включён в УСВК IBM LoadLeveler.

Большинство из рассмотренных выше алгоритмов ориентированы на классическую модель вычислительной системы, не учитывающей топологию вычислительного кластера и коммуникационные задержки при передаче данных. Существующие алгоритмы планирования, принимающие во внимание вычислительную сеть, допускают дальнейшее улучшение.

В рамках данной работы было выполнено экспериментальное исследование существующих списочных алгоритмов планирования задач и алгоритма Backfill с помощью симулятора вычислительного кластера и его управляющей системы [5]. В основу симулятора была положена классическая модель кластера и модель задачи с фиксированным количеством процессоров. Рассматривались сценарии гомогенного и гетерогенного кластера при наличии или отсутствии локальной загрузки вычислительных узлов.

Исследование показало, что лучшим является алгоритм Backfill, несколько хуже ведет себя алгоритм Most Processors First Served Scan. Однако, Backfill более сложен в реализации и требует наличия достаточно точных оценок времени выполнения задач. Для каждого сценария был рекомендован свой метод назначения задач и критерий доступности узлов для планирования.

В дальнейшем планируется доработка существующего программного симулятора вычислительного кластера с целью учета топологии вычислительной системы, многопроцессорности вычислительных узлов и коммуникационных задержек. Данный инструмент будет активно использоваться при разработке эффективных алгоритмов планирования и их сравнения с существующими вариантами – списочными и метаэвристическими алгоритмами.

Настоящая работа выполняется при поддержке Федерального агентства по образованию в рамках реализации ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009-2013 гг. (государственный контракт №П2039).

Список литературы

1. ***Blazewicz, J.*** *Handbook on Scheduling. From Theory to Applications / J. Blazewicz, K. Ecker, K. Pesch, G. Schmidt, J. Weglarz. – Берлин: Springer, 2007. – 647 с.*
2. ***Топорков, В.В.*** *Модели распределенных вычислений. – М.: ФИЗМАТЛИТ, 2004. – 320 с.*
3. ***Коваленко, В.Н.*** *Управление параллельными заданиями в гриде с неотчуждаемыми ресурсами / В.Н. Коваленко, Е.И. Коваленко, Д.А. Корягин, Д.А. Семячкин – Удаленный ресурс Института прикладной математики им. М.В. Келдыша Российской академии наук. – Режим доступа: http://www.keldysh.ru/papers/2007/source/prep2007_63.doc. – 34 с.*
4. ***Сальников, А.Н.*** *Система разработки и поддержки исполнения параллельных программ / А.Н. Сальников. – Диссертация на соискание ученой степени кандидата физико-математических наук. – 2006 г. – 92 с.*
5. ***Гергель, В.П.*** *Исследование алгоритмов планирования работ для высокопроизводительных кластерных систем / В.П. Гергель, М.Ю. Нестеренко, П.Н. Полежаев. – 8-я Международная Конференция по высокопроизводительным параллельным вычислениям на кластерных системах (HPC-2008). Казань, ноябрь 17-19, 2008. Труды конференции – Казань: Изд. КГТУ, 2008.*
6. ***Feitelson, D.*** *Utilization and predictability in scheduling the IBM SP2 with backfilling / D. Feitelson, A. Weil //Parallel Processing Symposium, 1998. IPPS/SPDP. – Орландо: Springer, 1998. – с. 542-546.*
7. ***Lawson, B.*** *Multiple-queue Backfilling Scheduling with Priorities and Reservations for Parallel Systems / B. Lawson, E. Smirni //Lecture Notes in Computer Science. – Берлин: Springer, 2002. – с. 72 – 87.*
8. ***Perkoviж, D.*** *Randomization, Speculation, and Adaptation in Batch Schedulers / D. Perkoviж, P. Keleher //Supercomputing, ACM/IEEE 2000 Conference Volume, Issue, 04-10 Nov. 2000. – с. 7 – 18.*
9. ***Костенко, В.А.*** *Исследование различных модификаций алгоритмов имитации отжига для решения задачи построения многопроцессорных расписаний / В.А. Костенко, А.В. Калашников. – Дискретные модели в теории управляющих систем. Труды VII Международной конференции. М.: МАКС Пресс, 2006 г. – с.179 – 184.*
10. ***Гаврилюк, А.Б.*** *Метод оптимального статического планирования задач в распределенных вычислительных системах с использованием генетического алгоритма / А.Б. Гаврилюк, В.А. Алексеев. – Проблемы программирования. 2004. № 1. – 2004 г. – с. 52 – 59.*
11. ***Abraham, A.*** *Nature's Heuristics for Scheduling Jobs on Computational Grids / A. Abraham, R. Buyya, B. Nath. – 2000 г. – Удаленный ресурс "The Cloud Computing and Distributed Systems (CLOUDS) Laboratory, University of Melbourne". – Режим доступа: www.cloudbus.org/papers/nhsjcg.pdf.*
12. ***Moore, M.*** *An Accurate and Efficient Parallel Genetic Algorithm to Schedule Tasks on a Cluster / M. Moore. – International Parallel and Distributed Processing Symposium (IPDPS'03). – 2003 г.*

*13. **Sinnen, O.** Task Scheduling for Parallel Systems. – Нью-Джерси, 2007. – 315 с.*