УДК 004.42
DOI: 10.17277/vestnik.2018.02.pp.228-235

PARALLEL HYBRID METAHEURISTICS APPROACH FOR OPTIMAL SELECTION OF PRODUCTION EQUIPMENT

A. B. Borisenko[1], S. Gorlatch[2]

Department of Computer-Integrated Systems in Mechanical Engineering,
TSTU (1), Tambov, Russia; borisenko@mail.gaps.tstu.ru;
Department of Mathematics and Computer Science,
University of Muenster (2), Germany

Keywords: combinatorial optimization; GPU computing; multiproduct batch plant; optimal equipment selection, parallel metaheuristics.

Abstract: In this paper we propose a parallel implementation of a hybrid metaheuristics-based algorithm on computers with Graphics Processing Units (GPU) using the CUDA programming environment. The presented hybrid algorithm combines two metaheuristics: Ant Colony Optimization (ACO) and Simulated Annealing (SA). We employ our approach for the real-world example application: optimal design of multi-product batch plants. We describe the parallelization of our algorithm, and we evaluate the performance of our GPU implementation. The results of our hybrid metaheuristic approach (ACO+SA) are very near to the global optimal solutions, and they are produced much faster than using the deterministic Branch-and-Bound approach.

Introduction

Selecting the production equipment of a Chemical-Engineering System (**CES**) is one of the main problems when designing chemical multi-product batch plants, e.g., for synthesizing chemical dyes and intermediate products, photographic materials, food, pharmaceuticals etc. Using multi-product batch plants offers the opportunity of quick response to market changes [1]. Building these plants from standardized modules can additionally help to reduce the time to market and costs.

The design problem consists in determining the number and capacity of the major processing equipment items, utilities, and storage tanks, such that the design and production objectives are met at the lowest possible capital and operating cost. The problem of optimal design of multi-product batch plants is typically formulated as a mixed integer nonlinear programming (**MINLP**) problem [2]. Many nonlinear models for batch design, which are based on the assumption of continuous sizes, can be reformulated as mixed integer linear programming (**MILP**) problems when sizes are restricted to discrete values [3]. Finding an optimal solution to this NP-hard problem can be a difficult task due to the so-called "combinatorial explosion" – the number of combinations to be examined grows exponentially, such that even the fastest computers require an intolerable amount of time to analyze them.

A metaheuristic is a generic algorithmic template that can find high-quality solutions of optimization problems [4] exploiting a trade-off of local search and global exploration. Metaheuristics usually finds good-quality solutions for optimization problems in a reasonable amount of time, but there is no guarantee that the optimal solution is always reached [5].

In this paper, we consider a challenging area of optimization – optimal design of multiproduct batch plants. There has been an active research on efficiently solving such and similar problems using metaheuristics. The classical n-queens problem was addressed using the Ant Colony Optimization (**ACO**) [6, 7] and its combination with a Genetic Algorithm (**GA**) [8].

In order to reduce the run time of metaheuristics-based approaches, their implementation on different parallel architectures has been studied. In particular, Graphics Processing Units (**GPUs**) are widely used by employing the CUDA platform [9].

Problem Formulation

A chemical-engineering system is a set of equipment (reactors, tanks, filters, dryers etc.) that implement the processing stages for manufacturing certain products. Assuming that the number of units at every stage of CES is fixed, the problem can be formulated as follows: CES consists of a sequence of I processing stages. Each i-th processing stage of the system can be equipped with equipment units from a finite set X_i, with J_i being the number of equipment units variants in X_i. All equipment unit variants of a CES are described as $X_i = \{x_{i,j}\}, i = \overline{1,I}, j = \overline{1,J_i}$, where $x_{i,j}$ is the main size j (also called working volume or working surface) of the unit suitable for processing stage i. Figure 1 shows an example of simple CES.

A chemical-engineering system variant $\Omega_e, e = \overline{1,E}$ of a CES, where $E = \Pi_{i=1}^{I} J_i$ is the number of all possible system variants, is an ordered set of equipment unit variants, selected from the respective sets.

Each variant Ω_e, of a system must be in an operable condition (expressed by the compatibility constraint) i.e., it must satisfy the conditions of a joint action for all its processing stages: $S(\Omega_e) = 0$ if the compatibility constraint is satisfied. An operable variant of a CES must run at a given production rate in a given period of time (processing time constraint), such that it satisfies the restrictions for the duration of its operating period $T(\Omega_e) \le T_{\max}$, where $T_{\max}$ is a given maximum period of time.

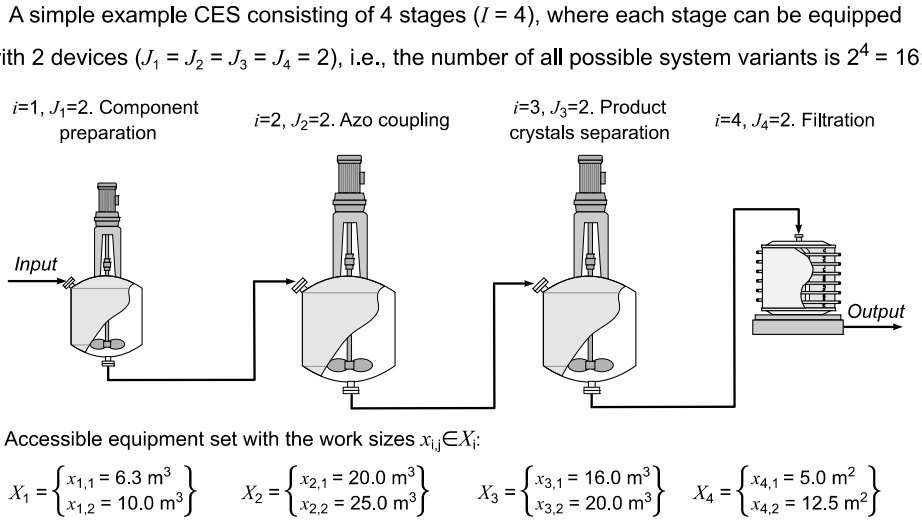

Fig. 1. Simple chemical-engineering system

Thus, designing an optimal CES can be formulated as the following optimization problem: to find a variant $\Omega_e^* \in \Omega_e, e = \overline{1,E}$ of a CES, where the optimality criterion – equipment costs $\mathrm{Cost}(\Omega_e^*)$ – reaches a minimum, and both the compatibility constraint and the processing time constraint are satisfied:

$$\Omega_e^* = \operatorname{argmin} \mathrm{Cost}(\Omega_e),\ e = \overline{1,E};$$
$$\Omega_e = \left\{x_{1,j_1}, x_{2,j_2}, ..., x_{I,j_I} \mid j_i = \overline{1,J_i}, i = \overline{1,I}\right\},\ e = \overline{1,E};$$
$$x_{i,j} \in X_i,\ i = \overline{1,I},\ j = \overline{1,J_i};$$
$$S(\Omega_e) = 0,\ e = \overline{1,E};$$
$$T(\Omega_e) \le T_{\max},\ e = \overline{1,E}.$$

We use the comprehensive mathematical model of CES operation, including expressions for checking constraints, calculating the optimization criterion, etc., which was initially presented in [10].

The Hybrid Metaheuristic Approach

Our approach to the optimal design of multi-product batch plants is based on two metaheuristics: Simulated Annealing (**SA**) and Ant Colony Optimization (ACO). SA is widely used for solving optimization problems [5]; its key advantage is escaping from local optima by allowing hill-climbing moves to find a global optimum. SA can deal with arbitrary systems and objective functions; it often finds an optimal solution and generally finds a good-quality solution.

Before searching for a solution using SA, we need a feasible initial solution. Our search for a feasible initial solution is a Constraint Satisfaction Problem (CSP) [11] without cost optimization, which consists in finding an operable variant of a CES. For solving it, we use the Ant Colony Optimization (ACO) metaheuristic that provides good-quality results in many applications, including CSP [7].

Figure 2 illustrates our parallel implementation of the hybrid (ACO+SA) approach on a system comprising a CPU and a GPU. The application code consists of a sequential code (host code for CPU) that invokes execution of hundreds or thousands of parallel threads on the device (GPU), where threads execute the kernel code.

The implementation proceeds in the following five steps (shown from left to right in Fig. 2):

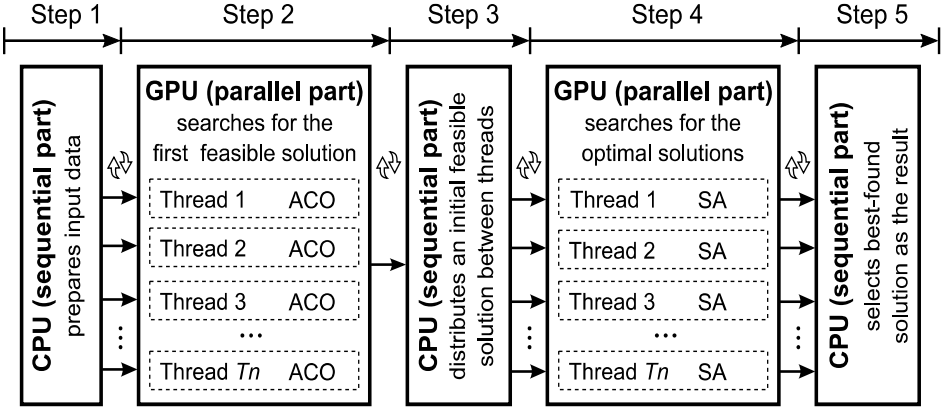

Fig. 2. Hybrid algorithm structure

1. CPU reads the input data, initializes the metaheuristics' parameters for ACO and SA, sends data to GPU, and starts the kernel for ACO on the GPU.

2. The ACO kernel on the GPU searches for the first feasible solution – the initial CES-variant. We use the Multiple Ant Colonies approach [12]: all colonies work as threads in parallel to solve the problem independently. If some thread finds a solution then all threads terminate. With more threads, the probability of finding a solution increases, and therefore the search time is reduced.

3. CPU receives the obtained feasible solution, distributes it between threads as an initial guess for SA, and starts the SA kernel function on the GPU.

4. The SA kernel on the GPU searches in each thread for the optimal solution with the initial solution found by ACO, i.e., we do not try to reduce the time of one iteration, but we rather increase the number of iterations executed simultaneously. Thus, the chance of the algorithm to converge to the global optimum increases, even if all instances use the same initial solution. A larger number of threads does not reduce the run time of the algorithm, but rather increases the probability that some thread eventually finds a nearly optimal solution.

5. CPU receives the SA solutions obtained by the GPU threads and chooses the best among them – this is the final solution of our problem.

Experimental Results

Our experiments are conducted on a hybrid system comprising: 1) a CPU: Intel Xeon E5-1620 v2, 4 cores with Hyper-Threading, 3.7 GHz with 16 GB RAM, and 2) a GPU: NVIDIA Tesla K20c with altogether 2496 CUDA cores and 5GB of global memory. We use Ubuntu 16.04.2, CUDA version 8.0, and GNU C++ Compiler version 5.4.0. The test CES example consist of 16 processing stages, with 2 to 12 variants of devices at every stage (total 2^{16} to 12^{16} CES variants). We solve the same problem on the same test system using our hybrid metaheuristic approach (ACO+SA), and we compare the results with the global optimal solution obtained by the Branch-and-Bound (B&B) approach.

Figure 3 shows the program run time (vertical axis has a logarithmic scale) of our hybrid approach vs. B&B depending on the problem size. The program run time of B&B

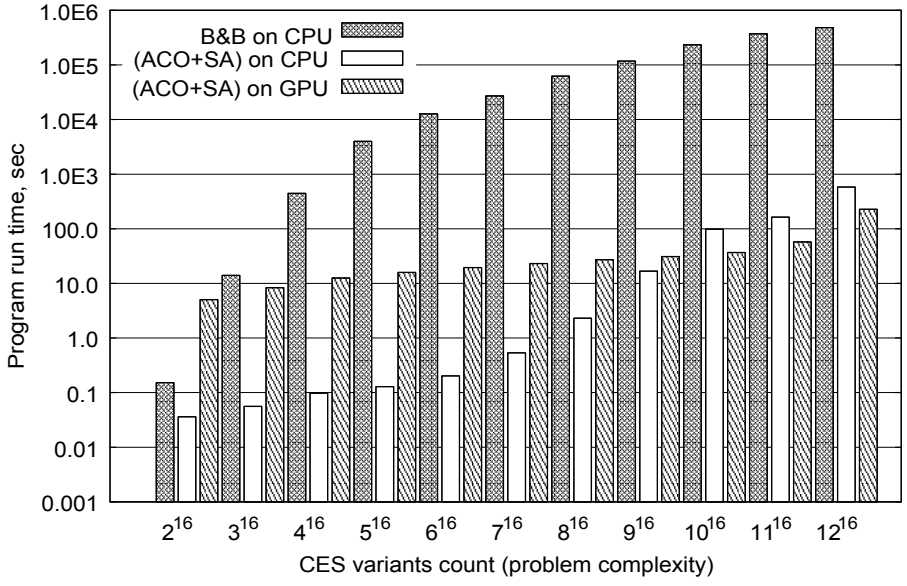

Fig. 3. Run time (logarithmic scale) depending on problem sizes

increases exponentially: while for problem size 2^{16} the run time is less than 1 sec, for size 12^{16} the run time of B&B becomes prohibitively long at about 134 hours. The run time of the sequential implementation of our hybrid algorithm is less than 0.1 sec for the smallest problem, and it increases to about 580 sec for our maximal problem size. The run time of the parallel implementation smoothly increases from 5 to 227 sec. The parallel implementation is slower than the sequential implementation for smaller problem sizes (2^{16} to 9^{16}), but for larger problem sizes (10^{16} to 12^{16}) it is faster than the sequential version by about 2.6 – 2.8 times. At first glance, the parallel speedup of 2.7 times compared to the sequential case is small, but the quality of the solutions obtained by the parallel implementation is significantly higher: the deviation from the global optimum for the parallel implementation is less than 0.01% against more than 10 % deviation for the sequential version. This good quality of solutions is achieved by the independent runs of SA: for a larger number of threads the probability that one of the threads finds a nearly optimal solution is higher; e.g., because running a parallel algorithm on 2500 threads is equivalent to the launch of the sequential algorithm 2500 times and the choice of the best among the found solutions.

Conclusion

Our new contribution in this paper is a novel hybrid (ACO+SA) metaheuristic approach to solving the optimization problem for multiproduct batch plants design and its parallel implementation on a CPU-GPU platform. We compare our results with the global optimal solution obtained by the B&B method. Our experiments confirm that our parallel hybrid approach obtains good-quality solutions which are very near to the global optimal values obtained by a deterministic algorithm like B&B, but our approach finds the solution much faster.

The research was generously supported by the DAAD (German Academic Exchange Service) and by the Ministry of Education and Science of the Russian Federation under "Mikhail Lomonosov II"-Programme.

References

1. Karpushkin S. V., Krasnyansky M.N., Borisenko A.B. Optimization of Existing Equipment for Multiproduct Batch Plants in New Product Release, *Transaction of the Tambov State Technical University*, 2016, Vol. 22, No. 2, P. 238-254.

2. Borisenko A., Haidl M., Gorlatch S. Using Parallel Branch-And-Bound Algorithm on GPUs For Optimal Design Of Multi-Product Batch Plants, *Transaction of the Tambov State Technical University*, 2015, Vol. 21, No. 3, P. 406-412.

3. Sandoval G. et al. Optimization of a Biotechnological Multiproduct Batch Plant Design for the Manufacture of Four Different Products: A Real Case Scenario, *Biotechnol. Bioeng*, 2017, Vol. 114, No. 6, P. 1252-1263.

4. Birattari M. Tuning Metaheuristics: A Machine Learning Perspective. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, 221 p.

5. Valadi J., Siarry P. Applications of Metaheuristics in Process Engineering / ed. Valadi J., Siarry P. Cham: Springer International Publishing, 2014, 444 p.

6. Khan S. et al. Solution of n-Queen problem using ACO // 2009 IEEE 13th International Multitopic Conference. IEEE, 2009, P. 1-5.

7. Solnon C. Ant Colony Optimization and Constraint Programming / ed. Solnon C., Jussien N. Hoboken, NJ: Wiley, 2013, 248 p.

8. Agarwal K., Sinha A., Hima Bindu M. A Novel Hybrid Approach to N-Queen Problem, *Advances in Computer Science, Engineering & Applications*, 2012, P. 519-527.

9. NVIDIA Corporation. CUDA C Programming Guide 9.1 [Electronic resource]. 2018. URL: http://docs.nvidia.com/cuda/pdf/CUDA_C_Programming_Guide.pdf.

10. Borisenko A.B., Karpushkin S.V. Hierarchy of Processing Equipment Configuration Design Problems for Multiproduct Chemical Plants, *Journal of Computer and Systems Sciences International*, 2014, Vol. 53, No. 3, P. 410-419.

11. Rossi F., Van Beek P., Walsh T. Handbook of Constraint Programming. Elsevier, 2006, 978 p.

12. Delévacq A. et al. Parallel Ant Colony Optimization on Graphics Processing Units, *Parallel and Distributed Computing*, 2013, Vol. 73, No. 1, P. 52-61.

Применение параллельного гибридного метаэвристического алгоритма для оптимального выбора технологического оборудования

А. Б. Борисенко[1], С. Горлач[2]

Кафедра «Компьютерно-интегрированные системы в машиностроении», ФГБОУ ВО «ТГТУ» (1); г. Тамбов, Россия; borisenko@mail.gaps.tstu.ru; Кафедра «Математика и информатика», Вестфальский университет имени Вильгельма (2), Мюнстер, Германия

Ключевые слова: комбинаторная оптимизация; метаэвристические алгоритмы; многоассортиментные производства; оптимальный выбор технологического оборудования; параллельные алгоритмы.

Аннотация: Представлена параллельная реализация параллельного гибридного метаэвристического алгоритма для вычислительных систем, оснащенных графическими ускорителями (Graphics Processing Units – GPU) с использованием среды программирования CUDA. Предлагаемый гибридный алгоритм объединяет две метаэвристики: алгоритм муравьиной колонии (Ant Colony Optimization – ACO) и алгоритм имитации отжига (Simulated Annealing – SA). Рассматриваемый подход используется для решения прикладной задачи – оптимального проектирования многоассортиментных химических производств. Проведены распараллеливание алгоритма и оценка его производительности на вычислительной системе, оснащенной GPU. Представленный гибридный (ACO+SA) подход позволяет получить решения, близкие к глобальным оптимальным, но за гораздо более короткое время по сравнению с детерминистическим алгоритмом, на основе ветвей-и-границ.

Список литературы

1. Карпушкин, С. В. Оптимизация существующего оборудования для многопрофильных паковых установок в выпуске новых продуктов / С. В. Карпушкин, М. Н. Краснянский, А. Б. Борисенко // Вестн. Тамб. гос. техн. ун-та. – 2016. – Т. 22, № 2. – С. 238 – 254.

2. Борисенко, А. Б. Использование параллельного ветвями-алгоритма на графических процессорах для оптимального проектирования многопродуктовых заготовок / А. Б. Борисенко, М. Хейдл, С. Горлач // Вестн. Тамб. гос. техн. ун-та. – 2015. – Т. 21, № 3. – С. 406 – 412.

3. Оптимизация разработки биотехнологического многопродуктового завода для производства четырех различных продуктов: сценарий реального случая / G. Sandoval [и др.] // Биотехнология. Bioeng. – 2017. Т. 114, № 6. – С. 1252 – 1263.

4. Бираттари, М. Настройка метаэвристики : перспектива машинного обучения / М. Бираттари. – Берлин : Springer Berlin Heidelberg, 2009. – 221 с.

5. Валади, Дж. Применение метаэвристики в технологической инженерии / Дж. Валади, П. Сирри. – Cham : Springer International Publishing, 2014. 444 с.

6. Решение проблемы n-Queen с использованием ACO / S. Khan [и др.] // 13-я Международная многоаспектная конференция. IEEE, 2009. – С. 1 – 5.

7. Solnon, C. Ant Colony Optimization and Constraint Programming / C. Solnon, N. Jussien. – Hoboken : J. Wiley, 2013. – 248 p.

8. Agarwal, K. Новый гибридный подход к проблеме N-Queen / K. Agarwal, A. Sinha, Bindu M. Hima // Достижения в области компьютерных наук, инженерии и приложений. – 2012. – С. 519 – 527.

9. Руководство по программированию CUDA C 9.1 [Электронный ресурс] / Корпорация NVIDIA. – Режим доступа : http://docs.nvidia.com/cuda/pdf/CUDA_C_Programming_Guide.pdf. (дата обращения 25.07.2018).

10. Борисенко, А. Б. Иерархия проблем проектирования конфигурации технологического оборудования для многопрофильных химических заводов / А. Б. Борисенко, С. В. Карпушкин // Journal of Computer and Systems Sciences International, 2014. – Т. 53, № 3. – С. 410 – 419.

11. Росси, Ф. Справочник о программировании ограничений / Ф. Росси, П. Ван Бик, Т. Уолш. Elsevier, 2006. – 978 с.

12. Параллельная оптимизация колоний на колонных модулях / A. Delévacq [и др.] // Parallel and Distributed Computing. – 2013. – Т. 73, № 1. – С. 52 – 61.

Anwendung des parallelen hybriden metaheuristischen Algorithmus zur optimalen Auswahl der technologischen Ausrüstung

Zusammenfassung: Es ist eine parallele Implementierung des parallelen hybriden metaheuristischen Algorithmus für Computersysteme vorgestellt, die mit Graphics Processing Units (GPUs) ausgestattet sind, die die CUDA Programmierumgebung verwenden. Der vorgeschlagene hybride Algorithmus kombiniert zwei Metaheuristiken: den Ant Colony Optimization (ACO) Algorithmus und den Simulated Annealing (SA) Algorithmus. Dieser Ansatz wird verwendet, um ein angewandtes Problem zu lösen – die optimale Projektierung der Mehrsortenchemikalienproduktion. Die Autoren parallelisierten den Algorithmus und bewerteten seine Leistung auf einem Computersystem, das mit einer GPU ausgestattet war. Der eingeführte hybride (ACO + SA) Ansatz ermöglicht es, Lösungen zu erhalten, die nahe dem globalen Optimum liegen, aber in einer viel kürzeren Zeit als im deterministischen Algorithmus, auf der Basis von Verzweigungen und Grenzen.

Application hybride parallèle de l'algorithme metaheuristique pour la sélection optimale de l’équipements technologique

Résumé: Est présentée la mise en œuvre parallèle d'un algorithme hybride parallèle metaheuristique pour les systèmes informatiques équipés d'accélérateurs graphiques (Graphics Processing Units – GPU) avec l’emploi de l'environnement de programmation CUDA. L'algorithme hybride proposé combine deux metaheuristiques: algorithme de la colone formique (AntColonyOptimization – ACO) et algorithme de simulation de recuit (Simulated Annealing – SA). Cette approche est utilisée pour

résoudre le problème appliqué – conception optimale de la production chimique multigamme. Les auteurs ont analysé l'algorithme et ont évalué ses performances à l'aide d'un système informatique équipé d'un GPU. L'approche hybride présentée (ACO+SA) permet d'obtenir des solutions proches à l'optimum global, mais en un temps beaucoup plus court que l'algorithme déterministe, basé sur les branches et les frontières.

Авторы: ***Борисенко Андрей Борисович*** – кандидат технических наук, доцент кафедры «Компьютерно-интегрированные системы в машиностроении», ФГБОУ ВО «ТГТУ», г. Тамбов, Россия; ***Горлач Сергей*** – PhD, профессор кафедры «Математика и информатика», руководитель группы «Параллельные и распределенные системы», Вестфальский университет имени Вильгельма, г. Мюнстер, Германия.

Рецензент: ***Егоров Сергей Яковлевич*** – доктор технических наук, профессор кафедры «Компьютерно-интегрированные системы в машиностроении», ФГБОУ ВО «ТГТУ», г. Тамбов, Россия.