Parallelization of combinatorial search when solving knapsack optimization problem on computing systems based on multi-core processors

P A Rahman

Department of Automated Technological and Informational Systems, Ufa State Petroleum Technological University, 2, October Avenue, Sterlitamak, 453118, Russia

E-mail: pavelar@yandex.ru

Abstract. This scientific paper deals with the model of the knapsack optimization problem and method of its solving based on directed combinatorial search in the boolean space. The offered by the author specialized mathematical model of decomposition of the search-zone to the separate search-spheres and the algorithm of distribution of the search-spheres to the different cores of the multi-core processor are also discussed. The paper also provides an example of decomposition of the search-zone to the several search-spheres and distribution of the search-spheres to the different cores of the quad-core processor. Finally, an offered by the author formula for estimation of the theoretical maximum of the computational acceleration, which can be achieved due to the parallelization of the search-zone to the search-spheres on the unlimited number of the processor cores, is also given.

1. Introduction

In the modern world, the combinatorial optimization problems [1] and methods of their solving become quite important in various science fields and spheres of human activity, such as computer science and engineering, economics and management, and in the many other areas, which deal with the finite sets of the boolean objects and parameters. In particular, the knapsack optimization problem [2] and its solving methods are often become a key part of the mathematical engine of the decision support systems [3, 4] in the modern enterprises.

In view of the fact that in most cases, the obtaining of the exact solution (global optimum) of the knapsack optimization problem is possible only through the full search, and it is impossible to provide the full search within the acceptable time for the large dimensions of the task, the approximate solving methods are often used, which are based on some limited directional search [5, 6]. Such solving methods provide limited combinatorial enumeration of the points in the \( n \)-dimensional space, and size of the enumeration is smaller than the size of the full search, but approximate solving methods do not guarantee the obtaining of the exact solution. However, the approximate solving methods at least can provide obtaining the suboptimal solution within the acceptable time.

In addition to the accuracy (optimality) of the solution, time of solution search is also quite important. Therefore, taking into account the large computing capabilities of the modern computing systems [7, 8] based on the multi-core processors [9, 10], the development of the efficient computational models providing fast parallelized combinatorial search during the solving process of the knapsack optimization problem is a quite urgent scientific problem.
In this scientific paper, the author offers the specialized model of decomposition of the search-zone to the separate search-spheres and the algorithm of distribution of the search-spheres to the different cores of the modern multi-core processors for the well-known local-search method, used as the approximate solving method for solving the knapsack optimization problem.

2. The mathematical model of the knapsack optimization problem and the method of its solving

At first, let us discuss the mathematical model of the knapsack optimization problem [1, 2].

A set of boolean variables $x_j \in \{0, 1\}, j = 1…n$, a constraint-inequality and an additive-weighted criterion function are given:

$$
\begin{align*}
    a_1x_1 + a_2x_2 + \ldots + a_nx_n & \leq b; \\
    L = c_1x_1 + c_2x_2 + \ldots + c_nx_n & \rightarrow \max; \\
    x_j & \in \{0, 1\}, j = 1…n.
\end{align*}
$$

Here, weight coefficients $c_j \geq 0, j = 1…n$ may take an arbitrary non-negative real value, and they have some kind of economical or technical meanings in the applied optimization tasks.

Coefficients $a_j \geq 0, j = 1…n$ and limit $b \geq 0$ may take an arbitrary non-negative real value. In fact, the given inequality defines some kind of the resource restriction (for example, restriction for the maximum summary weight, price, power-consumption and so on).

The aim of the optimization task is to find such set of values for the boolean variables, for which the given constraint is not violated and the criterion function takes the maximum possible value.

In present days, there is a set of methods for solving the optimization task described above, and all of them are based on some kind of the directed enumeration of the combinations of values of the boolean variables, where the combinations from the geometrical viewpoint can be interpreted as the points of the $n$-dimensional boolean space. In particular, the well-known local-search method [5, 6] provides search within the search-zones with given radius $r$. In the base variant of the local-search method, the radius is fixed and equal to one, and in the advanced variant, the radius may be more than one, and it can be given as an additional parameter for the local-search method.

The essence of the method is that beginning from some starting-point of the boolean space (usually, it is the point with the all zero coordinates), the method forms an $n$-dimensional search-zone with radius $r$ and centre in the starting-point, and then enumerates all points within the zone, by checking each point in the given constraint of the optimization task and calculating the value of the given criterion function. After completion of the enumeration of points in the zone, the method remembers the point for which the given constraint is not violated and the criterion function has the highest value, and this value is more than the value in the centre of the zone. Next, the method assumes this point as a new centre and forms a new $n$-dimensional search-zone with radius $r$, and then enumerates the points inside it. Such procedure is repeated inside until the last checked zone; there is no point, in which the criterion function has a better value than the value in the centre of the zone. In such case, the centre of the last zone will be accepted as solution of the optimization task.

Figure 1 shows the geometrical interpretation (projected to the two-dimensional plane) of the searching process through the search-zones, beginning from some starting point $X_0$ and ending in some final point $X_{opt}$.

**Figure 1.** Geometrical interpretation of the searching process in the local-search method.
3. Decomposition of the search-zone to the search-spheres

On the one hand, within the solving process of the knapsack optimization problem, the local-search method deals with the search-zones, and there is no way to provide a parallel searching process among the different search-zones, because each next search-zone is formed only after completion of the searching process in the previous search-zone.

On the other hand, inside each search-zone, it is possible to parallelize the combinatorial enumeration of the points inside the zone. Let us overview this aspect in detail.

At first, let us introduce some necessary mathematical definitions and statements.

An \( n \)-dimensional boolean space, which contains points \( X = (x_1,\ldots,x_n) \) with boolean coordinates \( x_j \in \{0,1\}, j = 1\ldots n \), is given.

**Definition 1.** Distance \( D(X_A, X_B) \) between two points \( X_A \) and \( X_B \) is defined on basis of the Hamming distance and is equal to the number of the unmatched boolean coordinates. Numerically, the distance can be calculated by the formula: \( D(X_A, X_B) = \sum_{j=1}^{n} |x_j^{(A)} - x_j^{(B)}| \).

**Statement 1.** The total number of all points in the \( n \)-dimensional boolean space is equal to \( 2^n \).

**Definition 2.** Sphere \( S_r(X_c) \) with given radius \( r \) and central point \( X_c \) is defined as a set of points, which have exact distance \( r \) from centre \( X_c \). In other words, \( D(X, X_c) = r, \forall X \in S_r(X_c) \).

**Statement 2.** The size of sphere \( S_r(X_c) \) is equal to the number of the points in it: \( |S_r(X_c)| = C^n_r \).

**Definition 3.** Zone \( V_r(X_c) \) with given radius \( r \) and central point \( X_c \) is defined as the set of all spheres with the radiuses from 0 to \( r \). In other words, \( \forall X \in V_r(X_c) \ \exists q \in [0, r]: X \in S_q(X_c) \) and, thus, \( V_r(X_c) = S_0(X_c) \cup S_1(X_c) \cup \ldots \cup S_r(X_c) \).

**Statement 3.** The size of zone \( V_r(X_c) \) is equal to the total number of the points in it: \( |V_r(X_c)| = C^n_0 + C^n_1 + \ldots + C^n_r = \sum_{q=0}^{r} C^n_q \).

Now, let us overview the following simple example of the search-zone for the given number of the boolean variables, \( n = 8 \), radius \( r = 5 \) and central point \( X_c = (0,0,0,0,0,0,0,0) \). It is possible to calculate easily the size of given zone \( V_5(X_c) \) using statement 3: \( \sum_{q=0}^{5} C^8_q = 219 \).

Search-zone \( V_5(X_c) \) by definition 3 consists of the search-spheres with radiuses 0, 1, 2, 3, 4 and 5, which contain in accordance with statement 2 the following number of points: \( C^0_8 = 1 \), \( C^1_8 = 8 \), \( C^2_8 = 28 \), \( C^3_8 = 56 \), \( C^4_8 = 70 \) and \( C^5_8 = 56 \). Thus, search-zone \( V_5(X_c) \) consists of:

- Sphere \( S_0(X_c) \), which contains 1 point: \([0,0,0,0,0,0,0,0]\).
- Sphere \( S_1(X_c) \), which contains 8 points: \([1,0,0,0,0,0,0,0],\ldots,[0,0,0,0,0,0,0,1]\).
- Sphere \( S_2(X_c) \), which contains 28 points: \([1,1,0,0,0,0,0,0],\ldots,[0,0,0,0,0,0,1,1]\).
- Sphere \( S_3(X_c) \), which contains 56 points: \([1,1,1,0,0,0,0,0],\ldots,[0,0,0,0,1,1,1,1]\).
- Sphere \( S_4(X_c) \), which contains 70 points: \([1,1,1,1,0,0,0,0],\ldots,[0,0,0,1,1,1,1,1]\).
- Sphere \( S_5(X_c) \), which contains 56 points: \([1,1,1,1,1,0,0,0],\ldots,[0,0,1,1,1,1,1,1]\).

Moreover, it is possible to consider the enumeration of the points in the zone as a set of the enumerations in the different spheres, which can be provided simultaneously and independently among them.

However, on the computing system with the one single-core processor, the decomposition of the search-zone to the set of the search-spheres does not provide any benefit, because the spheres, obviously, will be processed sequentially (figure 2).
4. Distribution of the search-spheres to the cores of the multi-core processors.

Now, let us note that the modern computers and computing systems can be based on microprocessors with four and more execution cores, and there is a way for parallelization of the combinatorial enumeration in the separate search-spheres via separate execution threads on the different cores.

In particular, in the described above example of the combinatorial enumeration, the distribution of the enumeration inside the different search-spheres on the four processor cores can be as follows:

- On processor core 0, 70 points of sphere $S_0(X_c)$ can be processed.
- On processor core 1, 56 points of sphere $S_3(X_c)$ can be processed.
- On processor core 2, 56 points of sphere $S_5(X_c)$ can be processed.
- On processor core 3, 1 point of sphere $S_0(X_c)$, 8 points of sphere $S_1(X_c)$ and 28 points of sphere $S_2(X_c)$ can be processed.

Thus, the total enumeration of the 219 points can be provided within the time equal to the time of enumeration of the 70 points of sphere $S_0(X_c)$ on processor core 0 in case of the parallelization of the enumeration relative to the four processor cores. The enumeration of the points in the other spheres on cores 1, 2 and 3 will be completed earlier.

Figure 3 shows a simplified picture of the parallelization of enumeration of points in the search-spheres relative to the four processor cores.

![Figure 3](image)

**Figure 3.** An example of the parallelization of enumeration of the points in the search-spheres.

Now, with taking into consideration all described above, one can offer a generalized mathematical model of the parallelization of combinatorial enumeration of the points in the search-spheres:
Here, $n$ – number of the boolean variables.

$m$ – number of the processor cores.

$V_r(X_c) –$ search-zone with radius $r$ in the $n$-dimensional boolean space, $r \leq n$.

$S_q(X_c) –$ search-sphere with radius $q$ as the part of the search-zone, $q = 0…r$.

$Q_k(X_c) –$ union of the spheres, assigned to the $k$-th processor core, $0 \leq k \leq m - 1$.

$J_k(X_c) –$ subset of radiuses of the spheres, assigned to the $k$-th processor core.

The key aspect of the mathematical model of the parallelization of combinatorial enumeration is that the distribution of set of search-spheres $S_q(X_c)$ to unions $Q_k(X_c)$, assigned to the different processor cores, should provide a minimal size of the largest union among all unions $Q_k(X_c)$. Such approach will give us the maximum possible computational acceleration.

Within the scientific research, the author also obtained a specialized algorithm for distribution of the search-spheres to the different processor cores (figure 4).

The algorithm starts with empty unions $Q_k(X_c) = \emptyset$, $k = 0…m - 1$, and set of search-spheres $T = \{S_0(X_c), S_1(X_c), …, S_r(X_c)\}$. Next, with each iteration, the largest search-sphere $S_{j^*}(X_c)$ in set $T$ and index $k^*$ of the processor core with the smallest total size of the assigned spheres are selected. After that, selected sphere $S_{j^*}(X_c)$ is removed from set $T$ and added to union $Q_{k^*}(X_c)$. Such procedure is repeated until set $T$ becomes empty. As the result, unions $Q_k(X_c)$, $k = 0…m - 1$, of the search-spheres, assigned to the different processor cores, are obtained.

In particular, for the discussed above example of the search-zone for the given number of the boolean variables, $n = 8$, and the radius of the search-zones, $r = 5$, one obtains the following variant of distribution of the search-spheres relative to the different processor cores for the number of the processor cores, $m = 4$, using the discussed above algorithm:
At last, it is also possible to estimate the theoretical maximum of the computational acceleration, which can be achieved due to the parallelization of the search-zone relative to the search-spheres on the unlimited number of the processor cores, by taking into the consideration that the largest search-sphere has size $c_n$, if $r < \left[\frac{n}{2}\right]$ and size $c_{n/2}$, if $r \geq \left[\frac{n}{2}\right]$:

$$K_{\text{max}} = \frac{1}{C_{\text{max}(r,(n/2))}} \sum_{q=0}^{r} C_{n,q}^{r}.$$  

Here, $\left[\frac{n}{2}\right]$ is the integer part of the division of $n$ by 2.

In particular, for the discussed above example for given $n = 8$ and $r = 5$, the theoretical maximum of computational acceleration $K_{\text{max}}$ equals $$\frac{1}{C_{8}^{5}} \sum_{q=0}^{5} C_{8,q}^{5} = \frac{219}{70} = 3.12.$$
Figure 4. An algorithm for distribution of the search-spheres relative to the different processor cores.

5. Conclusion
Thus, on the modern computing systems, based on the multi-core processors, it is possible to obtain good acceleration of the solving process of the knapsack optimization problem by using the specialized model of decomposition of the searching process to the separate independent subtasks and the algorithm of distribution of the subtasks to the different processor cores.

Within this scientific paper, the author overviewed the well-known mathematical model of the knapsack optimization problem and the solving method for it, and offered the specialized model of decomposition of the search-zone to the search-spheres and the algorithm of distribution of the search-spheres to the different cores of the computing system, based on the multi-core processors. An example of decomposition of the search-zone to the several search-spheres and distribution of the search-spheres to the different cores of the quad-core processor is also discussed.

At last, the author also offered a formula for estimation of the theoretical maximum of the computational acceleration, which can be achieved due to the parallelization of the search-zone to the search-spheres on the unlimited number of the processor cores.
Acknowledgements
The author is grateful to professor I. I. Ladygin from Moscow Power Engineering Institute for scientific support and theoretical base in the field of parallel and distributed computing.

References
[1] Cook W J, Cunningham W H, Pulleyblank W R and Schrijver A 1998 Combinatorial Optimization (New York: Wiley)
[2] Kellerer H, Pferschy U and Pisinger D 2004 Knapsack Problems (Springer-Verlag)
[3] Sharda R, Delen D and Turban E 2014 Business Intelligence and Analytics: Systems for Decision Support (Pearson Education)
[4] Povoa A P B, Corominas A and Luis de Miranda J 2017 Optimization and Decision Support Systems for Supply Chains (Springer International Publishing)
[5] Michiels W, Aarts E and Korst J 2007 Theoretical Aspects of Local Search (Springer-Verlag)
[6] Battiti R, Mauro B and Franco M 2008 Reactive Search and Intelligent Optimization (Springer-Verlag)
[7] Tanenbaum A S 2006 Structured Computer Organization (Prentice Hall PTR)
[8] Tanenbaum A S and Steen M 2007 Distributed Systems (Prentice Hall PTR)
[9] Keckler S W, Kunle O and Hofstee H P 2009 Multicore Processors and Systems (Springer-Verlag)
[10] Akhter S and Roberts J 2006 Multi-Core Programming: Increasing Performance through Software Multithreading (Intel Press)